Oracle® Application Server

Upgrade and Compatibility Guide

10*g* Release 3 (10.1.3)

Part No. B25585-01

January 2006

ORACLE®

Oracle Application Server Upgrade and Compatibility Guide, 10*g* Release 3 (10.1.3)

Part No. B25585-01

Primary Author: Peter LaQuerre

Contributing Author: Caroline Johnston

Contributors: Bryan Atsatt, Steve Button, Rachel Chan, Alfred Franci, Viresh Garg, Helen Grembowicz, Dan Hynes, Viral Kamdar, Kishore Kumar, Michael Lehmann, Rajiv Maheshwari, Robert Nettleton, Debu Panda, Paul Parkinson, Thomas Pfaeffle, Mary Beth Roeser, Ekkehard Rohwedder, J.J. Snyder, Jeff Steidl, Deborah Steiner, Jayaram Swaminathan, Sitaraman Swaminathan, Brian Wright

Contents

3 Redeploying J2EE Applications on 10*g* Release 3 (10.1.3)

4 Integrating 10*g* Release 3 (10.1.3) into Your Existing Environment

Preface

This preface contains the following sections:

Audience

This document is intended for Oracle Application Server 10*g* (9.0.4) and 10*g* Release 2 (10.1.2) administrators who want to:

- Understand the features and advantages of upgrading their Oracle Application Server J2EE environment to Oracle Application Server 10*g* Release 3 (10.1.3)
- Redeploy their J2EE applications in 10*g* Release 3 (10.1.3)
- Integrate 10*g* Release 3 (10.1.3) middle tiers into their existing 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) application server environment.
- Understand any compatibility issues between 10*g* (9.0.4), 10*g* Release 2 (10.1.2), and 10*g* Release 3 (10.1.3)

Documentation Accessibility

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

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

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

Related Documents

For more information, see the following documents in the Oracle Application Server 10*g* Release 3 (10.1.3) documentation library:

- *Oracle Application Server Installation Guide*
- *Oracle Application Server Administrator's Guide*
- *Oracle Containers for J2EE Configuration and Administration Guide*
- *Oracle Application Server Enterprise Deployment Guide*

Conventions

The following text conventions are used in this document:

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

1

Understanding the Upgrade to 10*g* Release 3 (10.1.3)

If you are already using Oracle Application Server 10*g* (9.0.4) or 10*g* Release 2 (10.1.2), this chapter introduces you to 10*g* Release 3 (10.1.3) and describes what is meant by an upgrade to 10*g* Release 3 (10.1.3).

This chapter contains the following sections:

1.1 What Is Oracle Application Server 10*g* Release 3 (10.1.3)?

Oracle Application Server 10*g* Release 3 (10.1.3) is a significant new release that provides a complete Java 2, Enterprise Edition (J2EE) 1.4-compliant environment. It does not include all the Oracle Application Server components that were available with 10*g* (9.0.4) and 10*g* Release 2 (10.1.2).

Instead, it is designed specifically for administrators who are using OC4J to deploy and manage J2EE applications.

Table 1–1 provides a summary of the components and features available with 10*g* Release 3 (10.1.3).

Table 1–1 Key Features of Oracle Application Server 10g Release 3 (10.1.3)

Feature Description	For More Information
A new version of Oracle Containers for J2EE (OC4J) provides the containers, APIs, and services mandated by the J2EE 1.4 specification.	*Oracle Containers for J2EE Configuration and Administration Guide*
A new version of the Application Server Control that is based on the Java Management Extensions (JMX) technology, including the J2EE Management and J2EE Application Deployment specifications.	"Introduction to Administration Tools" in the *Oracle Application Server Administrator's Guide*
A new version of Oracle Process Manager and Notification Server (OPMN), which provides clustering capabilities for Oracle Application Server10*g* Release 3 (10.1.3) instances.	Chapter 5, "Differences Between 10g Release 3 (10.1.3) and Previous Releases"

Table 1–1 (Cont.) Key Features of Oracle Application Server 10g Release 3 (10.1.3)

Feature Description	For More Information
The new version of OC4J also includes a new, more flexible method for grouping OC4J instances to facilitate the deployment and management of your J2EE applications across a cluster topology.	Chapter 5, "Differences Between 10g Release 3 (10.1.3) and Previous Releases"

1.2 What Does It Mean to Upgrade to 10*g* Release 3 (10.1.3)?

A typical upgrade to 10*g* Release 3 (10.1.3) involves the following steps:

1. Install your new 10*g* Release 3 (10.1.3) middle tiers.

   Note the following:

   - You can install a 10*g* Release 3 (10.1.3) middle tier on the same host as an existing 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) Oracle home, or you can install it on a separate host.
   - Use the installation options to create a 10*g* Release 3 (10.1.3) environment similar to your current 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) environment.

     For example, if you are using OracleAS Clusters in your current environment, install multiple 10*g* Release 3 (10.1.3) and create a 10*g* Release 3 (10.1.3) cluster.

     Note that 10*g* Release 3 (10.1.3) uses a new and improved clustering model. For more information, see Chapter 5, "Differences Between 10g Release 3 (10.1.3) and Previous Releases".

2. Optionally, apply any site-specific configuration settings to the new 10*g* Release 3 (10.1.3) Oracle HTTP Server and OC4J installations.

   For example, update the 10*g* Release 3 (10.1.3) Oracle HTTP Server `httpd.conf` file as necessary and define your data sources in the 10*g* Release 3 (10.1.3) OC4J instances.

3. Redeploy and test your 10*g* (9.0.4) and 10*g* Release 2 (10.1.2) J2EE applications on the new 10*g* Release 3 (10.1.3) environment.

   In some cases, this will require making changes to your applications to take advantage of new 10*g* Release 3 (10.1.3) capabilities and requirements.

4. When testing is complete, move your applications into the production in the new 10*g* Release 3 (10.1.3) environment.

5. Optionally, decommission the 10*g* (9.0.4) and 10*g* Release 2 (10.1.2) OC4J instances that were previously used to deploy and manage your J2EE applications.

> **See Also:** Chapter 2, "Step-By-Step Upgrade Examples" for detailed examples of upgrading from 10*g* Release 2 (10.1.2) to 10*g* Release 3 (10.1.3)

1.3 Documentation Road Map for Previous Oracle Application Server Users

Before you get started with Oracle Application Server 10*g* Release 3 (10.1.3), take advantage of the following documentation resources. For example, consider reviewing the 10*g* Release 3 (10.1.3) books in the following order:

1. Review the *Oracle Application Server Installation Guide*.

In particular, review Section 1.3, "Recommended Topologies," which describes common configurations to consider when installing 10*g* Release 3 (10.1.3).

2. Review the *Oracle Application Server Enterprise Deployment Guide*, which provides a detailed example of using 10*g* Release 3 (10.1.3) as the middle tier in an enterprise-wide deployment of Oracle Application Server.
3. Refer to the *Oracle Application Server Administrator's Guide* for information about the tools and procedures you can use to manage your 10*g* Release 3 (10.1.3) environment.
4. Refer to the *Oracle Containers for J2EE Configuration and Administration Guide* for general information about managing OC4J 10*g* Release 3 (10.1.3) instances.

2

Step-By-Step Upgrade Examples

This chapter provides detailed examples of upgrading from previous versions of Oracle Application Server to 10*g* Release 3 (10.1.3).

In each example, the FAQApp sample application is used to provide an example of how to redeploy an application on 10*g* Release 3 (10.1.3).

Refer to the following for more information:

- Upgrading FAQApp on a Single Oracle Application Server Instance
- Upgrading FAQApp in a Clustered Environment

2.1 Upgrading FAQApp on a Single Oracle Application Server Instance

The following sections provide an example of upgrading to 10*g* Release 3 (10.1.3) and redeploying an existing application on a single 10*g* Release 3 (10.1.3) OC4J instance:

- Starting Point for the FAQApp Upgrade
- Overview of the FAQApp Upgrade Procedure
- The FAQApp Upgrade Procedure

2.1.1 Starting Point for the FAQApp Upgrade

This procedure assumes the following starting point for the FAQApp upgrade:

- You have installed and configured a single 10*g* Release 2 (10.1.2) middle-tier Oracle home.

  Note: This procedure describes specifically how to upgrade from 10*g* Release 2 (10.1.2), but the procedure is also valid if you are using 10*g* (9.0.4) as your starting point.

- You have optionally made some site-specific changes to the Oracle HTTP Server configuration file (`httpd.conf`).
- You have successfully deployed the FAQApp sample application to the 10*g* Release 2 (10.1.2) middle tier.

  The FAQApp is available for download from the following location on Oracle Technology Network (OTN):

  `http://www.oracle.com/technology/tech/java/oc4j/demos/904/index.html`

See Also: "Configuring the FAQ Application Demo" in the *Oracle Application Server Containers for J2EE User's Guide* in the 10*g* Release 2 (10.1.2) documentation library, which is available from the following location on the Oracle Technology Network (OTN):

`http://www.oracle.com/technology/documentation/appserver.html`

- You now want to deploy the same FAQApp application on an instance of Oracle Application Server 10*g* Release 3 (10.1.3).
- You have installed and configured Apache Ant, which is a Java-based build tool that is included in the following directory of your 10*g* Release 3 (10.1.3) Oracle home:

  `(UNIX) 1013_ORACLE_HOME/ant/`
  `(Windows) 1013_ORACLE_HOME\ant\`

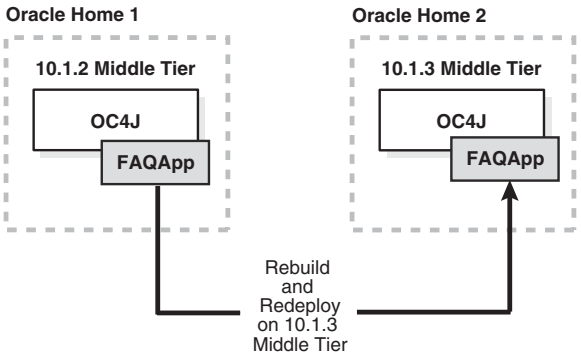
  Apache Ant, as well as documentation for using the tool, is also available from the following Web site:

  `http://ant.apache.org/`

2.1.2 Overview of the FAQApp Upgrade Procedure

The steps in the following procedure involve installing a new 10*g* Release 3 (10.1.3) Oracle home, modifying and rebuilding FAQApp, and then redeploying FAQapp on the new 10*g* Release 3 (10.1.3) OC4J instance, as shown in Figure 2–1.

Figure 2–1 Overview of the FAQApp Upgrade Procedure



2.1.3 The FAQApp Upgrade Procedure

The following sections describe the steps you can follow to install and configure a new 10*g* Release 3 (10.1.3) Oracle home and then recompile and redeploy the FAQApp on the new installation:

- Step 1: Install 10g Release 3 (10.1.3)
- Step 2: Apply Any Site-Specific Oracle HTTP Server Configuration Settings
- Step 3: Configure the FAQApp Data Source in the 10g Release 3 (10.1.3) Instance
- Step 4: Rebuild the FAQApp EAR File With Required JSP Library Archives
- Step 5: Deploy the Modified FAQApp EAR File on 10g Release 3 (10.1.3)
- Step 6: Test the FAQApp Sample Application on 10g Release 3 (10.1.3)

2.1.3.1 Step 1: Install 10*g* Release 3 (10.1.3)

Use the following steps to install a new 10*g* Release 3 (10.1.3) Oracle home. In this particular example, you will install a combined 10*g* Release 3 (10.1.3) Web server and OC4J instance in a single Oracle home:

1. Log in to the host computer and start Oracle Universal Installer.

   You can install the 10*g* Release 3 (10.1.3) Oracle home on the same host as the 10*g* Release 2 (10.1.2) J2EE and Web Cache Oracle home, or on a different host.

   > **See Also:** "Starting the Oracle Universal Installer" in the *Oracle Application Server Installation Guide*

2. On the Oracle Application Server 10g 10.1.3.0.0 Installation Screen, select **Basic Installation Mode** and select the options described in Table 2–1.

   This installation type installs a combined Web server and OC4J Instance in a single Oracle home.

   > **See Also:** 'Recommended Topologies" in the *Oracle Application Server Installation Guide* for more information about this and other 10*g* Release 3 (10.1.3) recommended topologies.

Table 2–1 Options to Select on the Installation Screen

Option	Description
Installation Directory	Enter the directory where you want install Oracle Application Server.
Basic Installation Mode	Select this option to install the combined Web server and OC4J instance in a single Oracle home.
Instance Name	The instance name identifies this Oracle Application Server instance. If you have more than one Oracle Application Server instance on the same host, the instances must have unique names.
Administration Username	The administration username for Oracle Application Server instances is set to `oc4jadmin` and cannot be changed. To manage Oracle Application Server instances using Oracle Enterprise Manager, log in as the `oc4jadmin` user.
Administration Password	Enter a password for the `oc4jadmin` user.
Confirm Password	Enter the password again to confirm that you entered it correctly.

3. Click **Install** and then follow the instructions on each screen to advance through the installation procedure.

   The installation is complete when the End of Installation Screen appears. This screen tells you whether or not your installation was successful. It also provides information about the default Oracle HTTP Server port, as well as the URL you can use to access the 10*g* Release 3 (10.1.3) Application Server Control Console.

2.1.3.2 Step 2: Apply Any Site-Specific Oracle HTTP Server Configuration Settings

If you have made any site-specific configuration changes to the Oracle HTTP Server `httpd.conf` configuration file in the 10*g* Release 2 (10.1.2) Oracle home, apply those changes to the `httpd.conf` file in the 10*g* Release 3 (10.1.3) Oracle home.

Like the 10*g* Release 2 (10.1.2) release, the 10*g* Release 3 (10.1.3) Oracle HTTP Server is based on the on Apache 1.3 Web Server. As a result, you can copy any configuration changes (such as modifications to the `httpd.conf` file) directly to the 10*g* Release 3 (10.1.3) Oracle home.

2.1.3.3 Step 3: Configure the FAQApp Data Source in the 10g Release 3 (10.1.3) Instance

The FAQApp uses a data source that consists of an Oracle database where the FAQ schema has been installed. This procedure assumes that you have successfully deployed and configured the FAQApp in a previous Oracle Application Server release. As a result, the FAQ schema already exists in an existing Oracle database.

However, before you deploy the FAQApp on the new 10*g* Release 3 (10.1.3) instance, use the following procedure to configure the `OracleDS` data source, which is required by the FAQApp application, in the new 10*g* Release 3 (10.1.3) OC4J container.

> **Note:** This procedure modifies the existing `OracleDS` datasource that is defined as part of the default application, which means that the datasource is available to all appliations desployed on this instance. Alternatively, you could define the data source using a `data-sources.xml` file within the FAQApp EAR file.
>
> For more information about defining JDBC data sources, see the *Oracle Containers for J2EE Services Guide*.

1. Use your browser to display the 10*g* Release 3 (10.1.3) Application Server Control Console.

   The URL for the Application Server Control Console is shown on the End of Installation Screen. This information is also saved to the following file in the 10*g* Release 3 (10.1.3) Oracle home:

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

   The first page that appears in the Application Server Control Console is the Cluster Topology page.

2. In the **Members** section of the Cluster Topology page, click **home** to display the OC4J Home page for the default `home` instance.

3. From the OC4J Home page, click **Administration**.

4. On the OC4J Administration page, click the task icon in the **JDBC Resources** row of the task table.

   Note that an "OracleDS" data source is created during the Oracle Application Server installation. However, this data source is a managed data source that relies on the "Example Connection Pool" to make its connections to the data base.

5. Click **Example Connection Pool** to display the Edit Connection Pool page.

6. Enter the JDBC URL that provides a connection to the Oracle database that hosts the FAQ schema and click **Apply**.

   Use the following format for the connection URL:

   ```
   jdbc:oracle:thin:@hostname:port:sid
   ```

   For example:

```
jdbc:oracle:thin:@appserv1.acme.com:1521:orcl
```

Hint: If you have trouble constructing the JDBC connection URL, you can optionally do the following:

1. Click **Create** in the Connection Pool section of the JDBC Resources page to create a new connection pool.
2. On the Connection Pool - Application page, select the default application, and then select **New Connection Pool**.
3. Click **Continue** to display the Create Connection Pool page. On this page, you can enter information about your database and Application Server Control will construct the JDBC URL for you automatically. From this page, you can also test the connection to be sure it is working before you proceed.
4. Delete the existing `OracleDS` data source and create a new `OracleDS` data source that uses the connection pool you just created.

7. Navigate to the Cluster Topology page and restart the OC4J instance to load the new connection URL for the JDBC connection pool.

   Note that when you restart the `home` instance, Application Server Control is also restarted. As a result, you must wait a few seconds and then enter the URL for the Application Server Control Console again. You can then log into the restarted Application Server Control Console.

2.1.3.4 Step 4: Rebuild the FAQApp EAR File With Required JSP Library Archives

The FAQApp requires the JavaServer Pages (JSP) Standard Tag Libraries. In previous versions of Oracle Application Server, these libraries were automatically available as part of the OC4J instance. In 10*g* Release 3 (10.1.3), if an application requires the JSP tag libraries, then these two jar files must be included as part of the application EAR file before you deploy the application.

> **See Also:** Section 3.2.2, "New Location for JavaServer Pages (JSP) Standard Tag Libraries (JSTL)"

Use the following procedure to locate the JSP Standard Tag Libraries in your existing 10*g* Release 3 (10.1.3) Oracle home and include them in the FAQApp EAR file:

1. If you have not done so already, unpack the `FAQApp.ear` file into a temporary, working directory.

   Note that a copy of the `FAQApp.ear` file that you deployed on 10*g* Release 2 (10.1.2) should be available in the following directory in the 10*g* Release 2 (10.1.2) Oracle home:

   ```
   (UNIX) 1012_ORACLE_HOME/j2ee/OC4J_Instance/applications/FAQApp/FAQApp.ear
   (Windows) 1012_ORACLE_HOME\j2ee\OC4J_Instance\applications\FAQApp\FAQApp.ear
   ```

   In this example, replace *1012_ORACLE_HOME* with the complete path to the 10*g* Release 2 (10.1.2) Oracle home, and replace *OC4J_Instance* with the name of the 10*g* Release 2 (10.1.2) OC4J instance that you used to deploy the FAQApp application.

2. Locate the following archives in the Oracle home of a 10*g* Release 3 (10.1.3) installation:

   UNIX:

   ```
   1013_ORACLE_HOME/j2ee/home/default-web-app/WEB-INF/lib/standard.jar
   1013_ORACLE_HOME/j2ee/home/default-web-app/WEB-INF/lib/jstl.jar
   ```

Windows:

```
1013_ORACLE_HOME\j2ee\home\default-web-app\WEB-INF\lib\standard.jar
1013_ORACLE_HOME\j2ee\home\default-web-app\WEB-INF\lib\jstl.jar
```

3. Copy these two .jar files to the following directory in the FAQApp working directory:

   ```
   faq/lib/
   ```

4. Delete any existing archives (.ear, .war, or other files) from the FAQApp working directory:

   ```
   faq/dist/
   ```

   Note that this step is important because the `ant all` command (which you run in the next step) will not remove or overwrite any existing archive files in the `dist` directory. As a result a new EAR file will not be created if a previous version of the EAR file exists in the `faq/dist` directory.

5. Use Apache Ant to rebuild the jar file using the following command:

   ```
   ant all
   ```

   When you enter this command, Apache Ant compiles and builds the FAQApp application EAR file, based on the information provided in the `build.xml` file that is included in the `faq` directory.

   See Also: Apache Ant Project Web Site at:

   ```
   http://ant.apache.org/
   ```

   Note: This example uses Apache Ant to rebuild the `FAQApp.ear` file. In fact, you can use other tools to perform this task. For more information, see the *Oracle Containers for J2EE Deployment Guide*.

6. Locate the new FAQApp EAR file in the following location in the FAQApp working directory:

   ```
   faq/dist/FAQApp.ear
   ```

2.1.3.5 Step 5: Deploy the Modified FAQApp EAR File on 10g Release 3 (10.1.3)

Use the following procedure to deploy the modified FAQApp sample application on your new 10*g* Release 3 (10.1.3) OC4J instance.

Note: This procedure describes how to use the Application Server Control Console to deploy the FAQApp sample application. Note, however, that OC4J 10*g* Release 3 (10.1.3) provides a number of options for deploying your applications.

For more information, see "Deployment Tool Options Provided with OC4J" in the *Oracle Containers for J2EE Deployment Guide*.

1. Navigate to the OC4J Home page in the 10*g* Release 3 (10.1.3) Application Server Control Console.
2. Click **Applications** to display the Applications page.

3. Click **Deploy**.
4. On the Deploy: Select Archive page, click **Browse** and select the modified `FAQApp.ear` file.

   Do not change the other default settings on the page.
5. On the Deploy: Application Attributes page, enter `FAQApp` in the **Application Name** field.

   Do not change the other default settings on the page.
6. On the Deploy: Deployment Settings page, click **Deploy** to deploy the FAQApp application on the 10*g* Release 3 (10.1.3) OC4J instance.

   Application Server Control displays the progress of the deployment and then displays a message indicating whether or not the deployment was successful.

   If the deployment was not successful, review the progress messages for any specific error messages.

   If any error messages mention SQL or database connections, the problem is likely with the data source configuration. Review the steps you performed in Section 2.1.3.3, "Step 3: Configure the FAQApp Data Source in the 10g Release 3 (10.1.3) Instance". Verify that the data source is valid and that the FAQ schema exists in the database. Verify that the database user name you used to connect to the database has accesss rights to the FAQ schema.

2.1.3.6 Step 6: Test the FAQApp Sample Application on 10*g* Release 3 (10.1.3)

If the deployment was successful, you can verify that the FAQApp application is working properly by accessing the following URL in your Web browser:

```
http://host.domain:port/FAQApp/
```

For example:

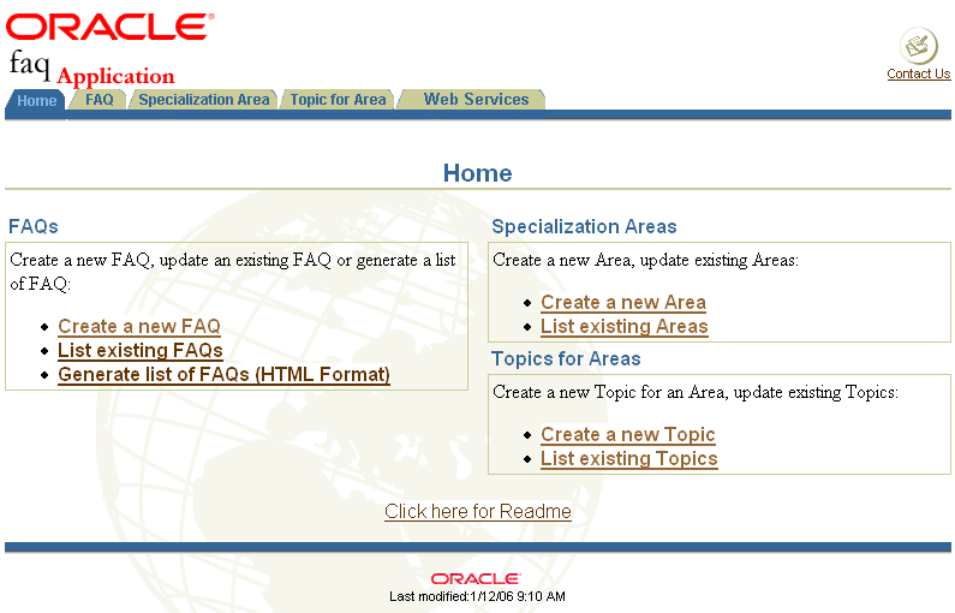
```
http://appserv1.acme.com:7779/FAQApp/
```

The FAQApp prompts you for a username and password. It is assumed you configured the users and roles for the application when you deployed FAQApp on your previous Oracle Application Server installation.

Enter `faq` as the username and `faq` as the password. The FAQApp home page should appear as shown in Figure 2–2. If the application does not display, note any error messages that appear in the browser window.

If any error messages mention SQL or database connections, the problem is likely with the data source configuration. Review the steps you performed in Section 2.1.3.3, "Step 3: Configure the FAQApp Data Source in the 10g Release 3 (10.1.3) Instance". Verify that the data source is valid and that the FAQ schema exists in the database. Verify that the database user name you used to connect to the database has accesss rights to the FAQ schema.

Figure 2–2 Successfully Deployed FAQApp Sample Application



2.2 Upgrading FAQApp in a Clustered Environment

The following sections provide an example of upgrading to a 10*g* Release 2 (10.1.2) clustered environment and redeploying an existing application on a new 10*g* Release 3 (10.1.3) cluster:

- Starting Point for Upgrading FAQApp in a Clustered Environment
- Overview of Upgrading FAQApp in a Clustered Environment
- The FAQApp Upgrade Procedure in a Clustered Environment

2.2.1 Starting Point for Upgrading FAQApp in a Clustered Environment

This procedure assumes the following starting point for the FAQApp upgrade:

- You have installed and configured a 10*g* Release 2 (10.1.2) OracleAS Farm and created an OracleAS Cluster within that farm.

  Note: This procedure describes specifically how to upgrade from 10*g* Release 2 (10.1.2), but the procedure is also valid if you are using 10*g* (9.0.4) as your starting point.

- You have successfully deployed the FAQApp sample application to the 10*g* Release 2 (10.1.2) OracleAS Cluster.

  The FAQApp is available for download from the following location on Oracle Technology Network (OTN):

  `http://www.oracle.com/technology/tech/java/oc4j/demos/904/index.html`

  See Also: "Configuring the FAQ Application Demo" in the *Oracle Application Server Containers for J2EE User's Guide* in the 10*g* Release 2 (10.1.2) documentation library

- You now want to deploy the same FAQApp application on a similar, Oracle Application Server 10*g* Release 3 (10.1.3) clustered environment.
- You have installed and configured Apache Ant, which is a Java-based build tool that is included in the following directory of your 10*g* Release 3 (10.1.3) Oracle home:

  ```
  (UNIX) 1013_ORACLE_HOME/ant/
  (Windows) 1013_ORACLE_HOME\ant\
  ```

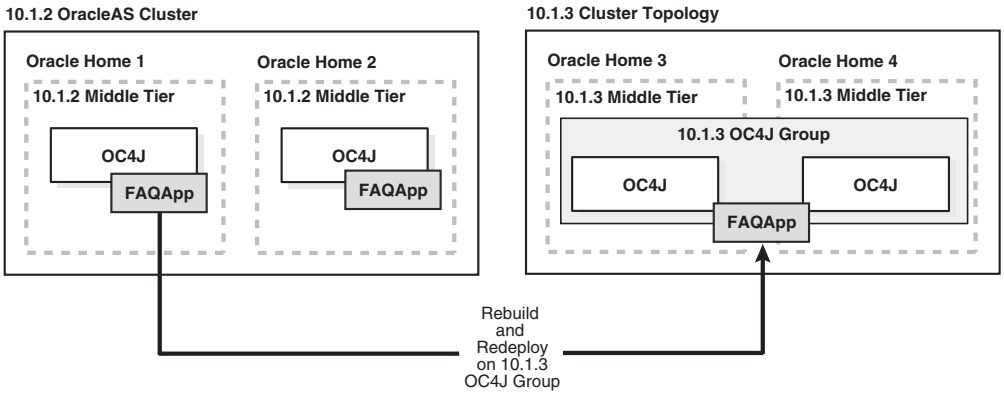
  Apache Ant, as well as documentation for using the tool, is also available from the following Web site:

  ```
  http://ant.apache.org/
  ```

2.2.2 Overview of Upgrading FAQApp in a Clustered Environment

The steps in the following procedure involve installing a new 10*g* Release 3 (10.1.3) Oracle home, modifying and rebuilding FAQApp, and then redeploying FAQapp on the new 10*g* Release 3 (10.1.3) OC4J instance, as shown in Figure 2–3.

Figure 2–3 Overview of Upgrading FAQApp in a Clustered Environment



2.2.3 The FAQApp Upgrade Procedure in a Clustered Environment

The following sections describe the steps you can follow to install and configure a 10*g* Release 3 (10.1.3) clustered environment and then redeploy the FAQApp on the new cluster:

- Step 1: Install and Configure a 10g Release 3 (10.1.3) Cluster
- Step 2: Apply Any Site-Specific Oracle HTTP Server Configuration Settings
- Step 3: Create an FAQApp Group
- Step 4: Configure the FAQApp Data Source for the Group
- Step 5: Rebuild the FAQApp EAR File With Required JSP Library Archives
- Step 6: Deploy FAQApp and Configure Application Clustering
- Step 7: Test the FAQApp Sample Application on 10g Release 3 (10.1.3)

2.2.3.1 Step 1: Install and Configure a 10*g* Release 3 (10.1.3) Cluster

Use the following steps to install a new 10*g* Release 3 (10.1.3) clustered environment:

Note: The following procedure describes how to install one of the recommended topologies, which are documented in more detail in the *Oracle Application Server Installation Guide*.

Specifically, this procedure describes how to install the topology described in the section, "Installing Multiple Combined Web Server and OC4J Instances to Form a Cluster" in the *Oracle Application Server Installation Guide*.

---

1. For the first middle tier Oracle home, install an Administration OC4J instance.

   After you start Oracle Universal Installer, select **Advanced Installation Mode** and then select **Integrated Web Server, J2EE Server and Process Management**.

   See Also: "Installing Integrated Web Server, J2EE Server and Process Management" in the *Oracle Application Server Installation Guide*

   During the installation procedure, follow the prompts, ensuring you do the following:

   - In the Administration Instance Settings page, select **Configure this as an Administration OC4J Instance**.
   - In the Administration Settings page, make a note of the `oc4jadmin` password you enter in the **Administrator Account Password** fields; you will need that password later.
   - In the Cluster Topology Configuration screen, select **Configure this HTTP Server instance to be part of an Oracle Application Server cluster topology** and specify a multicast discovery address for the cluster.

     Make a note of the address and port that you enter on this page; you will need it later.

     The multicast address you enter must be within the valid address range, which is `224.0.0.1` to `239.255.255.255`.

2. For the second middle tier, perform another **Integrated Web Server, J2EE Server and Process Management** advanced installation.

   See Also: "Installing Integrated Web Server, J2EE Server and Process Management" in the *Oracle Application Server Installation Guide*

   During the installation procedure, follow the prompts, ensuring you perform the following:

   - In the Administration Instance Settings page, *de*select **Configure this as an Administration OC4J Instance**.
   - In the Administration Settings page, enter the same `oc4jadmin` password in the **Administrator Account Password** fields that you entered in Step 1.

     It is important that you enter the same `oc4jadmin` password that you entered during the first installation. In order to use groups, each OC4J instance in the group must have the same `oc4jadmin` password. If they do not, then you will have to reset the password after the installation.

   - In the Cluster Topology Configuration screen, select **Configure this HTTP Server instance to be part of an Oracle Application Server cluster topology** and enter the same multicast address and port you entered in Step 1. Instances

that share the same multicast address are automatically configured as a cluster.

See Also: "Configuring Multiple OC4J Middle Tiers in a Cluster," in the *Oracle Application Server Administrator's Guide* for further information about configuring this topology

3. Verify that the installation and formation of the cluster was successful; do this by using your Web browser to display the 10*g* Release 3 (10.1.3) Application Server Control Console.

   The URL for the Application Server Control Console is shown on the End of Installation Screen. This information is also saved to the following file in the 10*g* Release 3 (10.1.3) Oracle home:

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

   When you first display the Application Server Control Console, both of the 10*g* Release 3 (10.1.3) installations should appear on the Cluster Topology page.

   See Also: Section 5.1, "Using the Cluster Topology Instead of an OracleAS Farm"

2.2.3.2 Step 2: Apply Any Site-Specific Oracle HTTP Server Configuration Settings

If you have made any site-specific configuration changes to the Oracle HTTP Server `httpd.conf` configuration file in the 10*g* Release 2 (10.1.2) OracleAS Cluster, apply those changes to the `httpd.conf` file in each of the 10*g* Release 3 (10.1.3) Oracle homes.

Like the 10*g* Release 2 (10.1.2) release, the 10*g* Release 3 (10.1.3) Oracle HTTP Server is based on the on Apache 1.3 Web Server. As a result, you can copy any configuration changes (such as modifications to the `httpd.conf` file) directly to the 10*g* Release 3 (10.1.3) Oracle homes.

See Also: *Oracle HTTP Server Administrator's Guide*

2.2.3.3 Step 3: Create an FAQApp Group

Within a 10*g* Release 3 (10.1.3) cluster, you can organize multiple OC4J instances into a single group. You can then perform specific deployment and configuration tasks on the group.

To create an group for the FAQApp:

1. Enter the following command in each of your new 10*g* Release 3 (10.1.3) Oracle homes:

   ```
   (UNIX) 1013_ORACLE_HOME/bin/createinstance -instanceName FAQApp
   (Windows) 1013_ORACLE_HOME\bin\createinstance -instanceName FAQApp
   ```

   Note the following important considerations when creating the new OC4J instances:

   - Be sure to use the same OC4J instance name (`FAQApp`) in each Oracle home. When you use the same name for two OC4J instances in your cluster (one in each Oracle home), Oracle Application Server automatically creates a group with that name.

- When prompted for the OC4J administrator's password, be sure to enter the same administrator password you used when you installed both 10*g* Release 3 (10.1.3) Oracle homes in Section 2.2.3.1; otherwise, you will not be able to perform group-wide operations.

2. In each 10*g* Release 3 (10.1.3) Oracle home, reload `opmn.xml` to load the new instance configuration, then restart OPMN to start the new instance. For example, on Windows:

```
1013_ORACLE_HOME\opmn\bin\opmnctl reload
1013_ORACLE_HOME\opmn\bin\opmnctl startall
```

3. Verify that the FAQApp OC4J instances were created, and that the FAQApp group was formed; do this by displaying the Cluster Topology page in the Application Server Control Console.

   A new FAQApp group should appear in the list of Groups on the Cluster Topology page. If the group does not appear, review the previous steps in this section to be sure you performed each step accurately.

2.2.3.4 Step 4: Configure the FAQApp Data Source for the Group

The FAQApp uses a data source that consists of an Oracle Database where the FAQ schema has been installed. This procedure assumes that you have successfully deployed and configured the FAQApp in a previous Oracle Application Server release. As a result, the FAQ schema already exists in an existing Oracle database.

However, before you deploy the FAQApp on the 10*g* Release 3 (10.1.3) group, you must use the following procedure to configure the `OracleDS` data source, which is required by the FAQApp application.

Use the following procedure to define the data source for all the OC4J instances in the FAQApp group:

1. From the Cluster Topology page in the Application Server Control Console, scroll to the Groups section of the page.
2. Click the name of the FAQApp group.
3. From the Group page, click **Administration**.
4. On the Group Administration page, click the task icon in the **JDBC Resources** row of the task table.

   Note that an "OracleDS" data source has already been created. However, this data source is a managed data source that relies on the "Example Connection Pool" to make its connections to the data base.

5. Click **Example Connection Pool** to display the Edit Connection Pool page.
6. Enter the JDBC URL that provides a connection to the Oracle database that hosts the FAQ schema and click **Apply**.

   Use the following format for the connection URL:

   `jdbc:oracle:thin:@`*`hostname`*`:`*`port`*`:`*`sid`*

   For example:

   `jdbc:oracle:thin:@appserv1.acme.com:1521:orcl`

Hint: If you have trouble constructing the JDBC connection URL, you can optionally do the following:

1. Click **Create** in the Connection Pool section of the JDBC Resources page to create a new connection pool.
2. On the Connection Pool - Application page, select the default application, and then select **New Connection Pool**.
3. Click **Continue** to display the Create Connection Pool page. On this page, you can enter information about your database and Application Server Control will construct the JDBC URL for you automatically. From this page, you can also test the connection to be sure it is working before you proceed.
4. Delete the existing OracleDS data source and create a new `OracleDS` data source that uses the connection pool you just created.

7. Restart the FAQApp group to load the new JDBC connection pool URL:
   a. Navigate to the Cluster Topology page.
   b. Select the FAQApp group.
   c. Click **Stop**.
   d. After Application Server Control confirms that the group has been stopped, select the FAQApp group again and click **Start**.

2.2.3.5 Step 5: Rebuild the FAQApp EAR File With Required JSP Library Archives

The FAQApp requires the JavaServer Pages (JSP) Standard Tag Libraries. In previous versions of Oracle Application Server, these libraries were automatically available as part of the OC4J instance. In 10*g* Release 3 (10.1.3), if an application requires the JSP tag libraries, then these two jar files must be included as part of the application EAR file before you deploy the application.

> **See Also:** Section 3.2.2, "New Location for JavaServer Pages (JSP) Standard Tag Libraries (JSTL)"

Use the following procedure to locate the JSP Standard Tag Libraries in your existing 10*g* Release 3 (10.1.3) Oracle home and include them in the FAQApp EAR file:

1. If you have not done so already, unpack the `FAQApp.ear` file into a temporary, working directory.

   Note that a copy of the FAQApp.ear file that you deployed on 10*g* Release 2 (10.1.2) should be available in the following directory in the 10*g* Release 2 (10.1.2) Oracle home:

   ```
   (UNIX) 1012_ORACLE_HOME/j2ee/OC4J_Instance/applications/FAQApp/FAQApp.ear
   (Windows) 1012_ORACLE_HOME\j2ee\OC4J_Instance\applications\FAQApp\FAQApp.ear
   ```

   In this example, replace *1012_ORACLE_HOME* with the complete path to the 10*g* Release 2 (10.1.2) Oracle home, and replace *OC4J_Instance* with the name of the OC4J instance that you used to deploy the FAQApp application.

2. Locate the following archives in the Oracle home of a 10*g* Release 3 (10.1.3) installation:

   UNIX:

   ```
   1013_ORACLE_HOME/j2ee/home/default-web-app/WEB-INF/lib/standard.jar
   1013_ORACLE_HOME/j2ee/home/default-web-app/WEB-INF/lib/jstl.jar
   ```

Windows:

```
1013_ORACLE_HOME\j2ee\home\default-web-app\WEB-INF\lib\standard.jar
1013_ORACLE_HOME\j2ee\home\default-web-app\WEB-INF\lib\jstl.jar
```

3. Copy these two .jar files to the following directory in the FAQApp working directory:

   ```
   faq/lib/
   ```

4. Delete any existing archives (.ear, .war, or other files) from the FAQApp working directory:

   ```
   faq/dist/
   ```

   Note that this step is important because the `ant all` command (which will be used in the next step) will not remove or overwrite any existing archive files in the dist directory. As a result a new EAR file will not be created if a previous version of the EAR file exists in the `faq/dist` directory.

5. Use Apache Ant to rebuild the jar file using the following command:

   ```
   ant all
   ```

   When you enter this command, Apache Ant compiles and builds the FAQApp application EAR file, based on the information provided in the build.xml file that is included in the `faq` directory.

   See Also: Apache Ant Project Web Site at:

   ```
   http://ant.apache.org/
   ```

   Note: This example uses Apache Ant to rebuild the `FAQApp.ear` file. In fact, you can use other tools to perform this task. For more information, see the *Oracle Containers for J2EE Deployment Guide*.

6. Locate the new FAQApp EAR file in the following location in the FAQApp working directory:

   ```
   faq/dist/FAQApp.ear
   ```

2.2.3.6 Step 6: Deploy FAQApp and Configure Application Clustering

Use the following procedure to deploy the modified FAQApp sample application on the 10*g* Release 3 (10.1.3) FAQApp group.

During the deployment, you can also configure application clustering for the FAQApp, which enables session state replication across the application instances in the cluster. Examples of session state information include whether or not a user is logged in and the contents of a shopping cart.

Oracle Application Server 10*g* Release 3 (10.1.3) offers three protocols for session state replication. In this example, you enable the peer-to-peer replication protocol. When you select this protocol, OPMN automatically replicates state information to the other OC4J instances in the 10*g* Release 3 (10.1.3) cluster.

See Also: Section 5.3, "Using Application Clustering for State Replication"

To deploy the FAQApp on the FAQApp group and configure application clustering:

1. Navigate to the Cluster Topology page in the 10*g* Release 3 (10.1.3) Application Server Control Console.
2. Click the name of the FAQApp group to display the FAQApp Group page.
3. Click **Applications** to display the Group Applications page.
4. Click **Deploy**.
5. On the Deploy: Select Archive page, click **Browse** and select the updated `FAQApp.ear` file.

   Do not change the other default settings on the page.
6. On the Deploy: Application Attributes page, enter `FAQApp` in the **Application Name** field.

   Do not change the other default settings on the page.
7. On the Deploy: Deployment Settings page, click **Configure Clustering** and use the resulting page to enable session state replication for the FAQApp application:
   a. Select **Override parent application clustering settings**.
   b. Select **Enable** from the **Clustering** drop-down menu.
   c. Select **Peer-Peer Replication**; do not enter a value in the **Bind Address** field unless the host computer has multiple network cards.
   d. Click **OK**.
8. Click **Deploy** to deploy FAQApp to both OC4J instances in the FAQApp group.

   Application Server Control displays the progress of the deployment and then displays a message indicating whether or not the deployment was successful.

   If the deployment was not successful, review the progress messages for any specific error messages.

   If any error messages mention SQL or database connections, the problem is likely with the data source configuration. Review the steps you performed in Section 2.2.3.4, "Step 4: Configure the FAQApp Data Source for the Group". Verify that the data source is valid and that the FAQ schema exists in the database. Verify that the database user name you used to connect to the database has accesss rights to the FAQ schema.

2.2.3.7 Step 7: Test the FAQApp Sample Application on 10*g* Release 3 (10.1.3)

If the deployment was successful, you can verify that the FAQApp application is working properly by accessing the following URL in your Web browser:

`http://`*`host.domain`*`:`*`port`*`/FAQApp/`

For example:

```
http://appserv1.acme.com:7779/FAQApp/
```

The FAQApp prompts you for a username and password. It is assumed you configured the users and roles for the application when you deployed FAQApp on your previous Oracle Application Server installation.

Enter `faq` as the username and `faq` as the password. The FAQApp home page should appear as shown in Figure 2–2. If the application does not display, note any error messages that appear in the browser window.

If any error messages mention SQL or database connections, the problem is likely with the data source configuration. Review the steps you performed in Section 2.2.3.4, "Step 4: Configure the FAQApp Data Source for the Group". Verify that the data source is valid and that the FAQ schema exists in the database. Verify that the database user name you used to connect to the database has accesss rights to the FAQ schema.

3

Redeploying J2EE Applications on 10*g* Release 3 (10.1.3)

Oracle Application Server 10*g* Release 3 (10.1.3) introduces support for the the latest J2EE 1.4 technologies and APIs. As a result, you can use this release to deploy J2EE applications that take advantage of the newest J2EE features and capabilities.

This chapter provides important considerations to review before you deploy your Oracle Application Server 10*g* (9.0.4) and 10*g* Release 2 (10.1.2) applications on Oracle Application Server 10*g* Release 3 (10.1.3).

This chapter includes the following sections:

3.1 Overview of Redeploying Applications on 10*g* Release 3 (10.1.3)

Oracle Application Server 10*g* Release 3 (10.1.3) supports functionality outlined in the J2EE Application Deployment API (JSR-88), which defines a standard API for configuring and deploying J2EE applications and modules into a J2EE-compatible environment.

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

- Start an application immediately upon deployment, making it available to clients
- Stop an application, making it unavailable to clients
- Undeploy an application or module
- Redeploy an application or module, essentially updating the currently installed application with an updated version
- Create a deployment plan containing the aggregated OC4J-specific configuration data needed to deploy a component into OC4J.

> **See Also:** "Working With Deployment Plans" in the *Oracle Containers for J2EE Deployment Guide* for details on the JSR-88 implementation in OC4J.

To deploy an application, you use one of two management tools:

- The new Application Server Control Console provided with 10*g* Release 3 (10.1.3)

  > **See Also:** "Introduction to Administration Tools" in the *Oracle Application Server Administrator's Guide*

- The `admin_client.jar` command-line utility, which is new for 10*g* Release 3 (10.1.3)

  > **See Also:** Section 5.5, "Using the admin_client.jar Utility to Manage OC4J Instances and Clusters"

For complete information about deploying your J2EE applications on 10*g* Release 3 (10.1.3), see the *Oracle Containers for J2EE Deployment Guide*.

For a step-by-step example of upgrading to 10*g* Release 3 (10.1.3) and redeploying the FAQApp sample application on 10*g* Release 3 (10.1.3), see Chapter 2, "Step-By-Step Upgrade Examples".

3.2 General Considerations

The following sections describe general information that you should consider before redeploying your applications on 10*g* Release 3 (10.1.3):

- Classloading and Shared Library Support
- New Location for JavaServer Pages (JSP) Standard Tag Libraries (JSTL)
- Oracle JSP Markup Language (JML) Tag Library No Longer Supported

3.2.1 Classloading and Shared Library Support

Oracle Application Server 10*g* Release 3 (10.1.3) offers significant improvements in the areas of class loading and shared library support.

For complete information about how you can take advantage of these new features, see "Utilizing the OC4J Class Loading Framework" in the *Oracle Containers for J2EE Developer's Guide*.

That chapter in the *Oracle Containers for J2EE Developer's Guide* contains an overview of the new class loading framework, information about using shared libraries, as well as classloading best practices and troubleshooting information.

3.2.2 New Location for JavaServer Pages (JSP) Standard Tag Libraries (JSTL)

In previous versions of Oracle Application Server, the JavaServer Pages (JSP) Standard Tag Libraries were automatically available as part of the OC4J instance. However, application developers often want to include their own custom version of the libraries, or a newer version of the tag libraries. In previous versions of OC4J, errors could result if you included custom tag libraries in addition to the pre-packaged libraries.

As a result, for 10*g* Release 3 (10.1.3), the tag libraries (`standard.jar` and `jstl.jar`) are now installed in a new location in the Oracle Application Server

Oracle home. If your application depends upon these libraries, you must now include the tag libraries in the `WEB-INF/lib` directory of your application EAR file.

Specifically, the libraries are now installed in the following directory. You can copy these libraries from this location and include them into your application before you deploy the application on 10*g* Release 3 (10.1.3):

1013_ORACLE_HOME`/j2ee/home/default-web-app/WEB-INF/lib`

> **See Also:** "Support for the JavaServer Pages Standard Tag Library" in the *Oracle Containers for J2EE JSP Tag Libraries and Utilities Reference*

3.2.3 Oracle JSP Markup Language (JML) Tag Library No Longer Supported

The Oracle JSP Markup Language (JML) tag library is officially de-supported as of Oracle Application Server 10*g* Release 3 (10.1.3).

Developers are advised to use tags provided with the JavaServer Pages Standard Tag Library (JSTL), which provide similar functionality in a standardized implementation.

For more information, see Chapter 2, "Support for the JavaServer Pages Standard Tag Library" in the *Oracle Containers for J2EE Support for JavaServer Pages Developer's Guide*.

3.3 Data Source Considerations

The following sections provide information about using data sources in 10*g* Release 3 (10.1.3):

- New Features for Data Sources in 10g Release 3 (10.1.3)
- Converting data-sources.xml to the New 10g Release 3 (10.1.3) Format
- Using Oracle JDBC-OCI Drivers with 10g Release 3 (10.1.3)

3.3.1 New Features for Data Sources in 10*g* Release 3 (10.1.3)

The following OC4J Data Source features and behaviors are new for this release:

- Data source configuration can be performed entirely in the Oracle Enterprise Manager 10g Application Server Control Console.
- The OC4J Data Source types are **managed data sources** and **native data sources**, replacing emulated, non-emulated, and native.
- New connection caching mechanism that is uniform across Oracle data sources and offers integrated Real Application Clusters (RAC) failover support.

> **See Also:** "Data Sources" in the *Oracle Containers for J2EE Services Guide*
>
> "Managing Data Sources and JDBC Connection Pools" in the Application Server Control online help

3.3.2 Converting data-sources.xml to the New 10*g* Release 3 (10.1.3) Format

Oracle Application Server 10*g* Release 3 (10.1.3) introduces a new format for the `data-sources.xml` file, which defines the data sources for your application, OC4J instance, or group.

However, you can still use your existing `data-source.xml` files. OC4J will convert the data sources to the new format at runtime. Note, however, that if you deploy an EAR file that contains a data-sources.xml file in the previous format, OC4J will convert

the `data-sources.xml` file that is expanded on disk. It will not modify the `data-sources.xml` file contained within the EAR file.

Alternatively, if you are using standalone OC4J, you can use the `admin.jar` utility to convert the `data-sources.xml` file to the new format.

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

For more information, see "Converting Existing Data Sources to the New Configuration" in the *Oracle Containers for J2EE Configuration and Administration Guide*.

3.3.3 Using Oracle JDBC-OCI Drivers with 10*g* Release 3 (10.1.3)

If your existing applications use the Oracle JDBC Oracle Call Interface (OCI) driver, be sure to review the section, "Oracle JDBC Drivers" in the *Oracle Containers for J2EE Services Guide* for information on configuration and upgrade requirements.

3.4 Web Services Considerations

For backward compatibility, Oracle Application Server 10*g* Release 3 (10.1.3) includes the underlying software required to run 10*g* Release 2 (10.1.2) Web services. As a result, Web services applications designed and packaged to run with Oracle Application Server 10*g* (9.0.4) and 10*g* Release 2 (10.1.2) can be used without modification with Release 3.

However, there are significant advantages to recreating your Web services for 10*g* Release 3 (10.1.3). For complete information on creating Web services for 10*g* Release 3 (10.1.3), refer to the *Oracle Application Server Web Services Developer's Guide*.

In addition, refer to the following sections for specific considerations when using 10*g* Release 3 (10.1.3) to recreate Web services that were originally created against 10*g* (9.0.4) or 10*g* Release 2 (10.1.2):

- New Web Services Assembler (wsa.jar)
- Assembling Web Services From Java Classes in 10g Release 3 (10.1.3)
- Developing Database Web Services in 10g Release 3 (10.1.3)

3.4.1 New Web Services Assembler (wsa.jar)

If you re-create your Web services for the 10*g* Release 3 (10.1.3), note that the Web Services Assembler tool for 10*g* Release 3 (10.1.3) is now called `wsa.jar`, and it is not compatible with the Web Services Assembler tool used for previous releases (`WebServicesAssembler.jar`). Web services and clients created with `wsa.jar` will be different and incompatible with Web services created with `WebServicesAssembler.jar`.

> **See Also:** "Using WebServicesAssembler" in the *Oracle Application Server Web Services Developer's Guide*

3.4.2 Assembling Web Services From Java Classes in 10*g* Release 3 (10.1.3)

If you created a Web service based on a Java class for Oracle Application Server 10*g* (9.0.4) or 10*g* Release 2 (10.1.2), you can do the same using the new Web Services

Assembler (`wsa.jar`) available with 10*g* Release 3 (10.1.3). However, you must be aware of the following:

- In 10*g* Release 2 (10.1.2), it was possible to publish a class by itself without providing an interface. In 10*g* Release 3 (10.1.3), you must provide an interface (specifically, the Service Endpoint Interface) to publish a class.

  See Also: "Writing Java Class-Based Web Services" in the *Oracle Application Server Web Services Developer's Guide*

- The set of Java types that are natively supported has changed with the 10*g* Release 3 (10.1.3) release. For a list of the supported data types, see the JAX-RPC 1.1 specification available from the following URL:

  `http://java.sun.com/xml/jaxrpc/index.jsp`

  See Also: "Assembling a Web Service with Java Classes" in the *Oracle Application Server Web Services Developer's Guide*

3.4.3 Developing Database Web Services in 10*g* Release 3 (10.1.3)

If you created database Web services with Oracle Application Server 10*g* (9.0.4) or 10*g* Release 2 (10.1.2), you can do the same using the new Web Services Assembler (`wsa.jar`) available with 10*g* Release 3 (10.1.3).

Note, however, that in 10*g* (9.0.4) and in 10*g* Release 2 (10.1.2), database Web services were always created using the RPC-encoded message format. In 10*g* Release 3 (10.1.3), database Web services are by default created using the document-literal message format.

See Also: "Supported Message Formats" in the *Oracle Application Server Web Services Developer's Guide*

As a result, if you use the RPC-encoded message format when you create a 10*g* Release 3 (10.1.3) database Web service, the Web service will not be interchangeable between 10*g* Release 3 (10.1.3) and previous Oracle Application Server Web services clients.

Specifically, a Web service client written for a database Web service generated under 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) will fail if you try to use it against a database Web service generated under 10*g* Release 3 (10.1.3). This will be true even if the PL/SQL structures have remained the same.

One of the reasons for this is that the SQL collection type was mapped into a complex type with a single array property in 10*g* (9.0.4) and 10*g* Release 2 (10.1.2). In release 10*g* Release 3 (10.1.3), it is mapped directly into an array instead.

If you regenerate the Web service client, you will have to rewrite the client code. This is because the regenerated code will now be employing an `array[]` instead of a `BeanWrappingArray`.

See Also: "Developing Database Web Services" in the *Oracle Application Server Web Services Developer's Guide*

3.5 Java Messaging Service (JMS) Considerations

The following sections provide information about using JMS 10*g* Release 3 (10.1.3):

- Nomenclature Changes for 10g Release 3 (10.1.3) JMS Support
- Using the JMS Connector Provided by 10g Release 3 (10.1.3)

3.5.1 Nomenclature Changes for 10g Release 3 (10.1.3) JMS Support

In past releases, Oracle used the terms "OracleAS JMS" and "OJMS" when describing the In-Memory, File-Based, and Database persistence options. "OracleAS JMS" referred to the In-Memory and File-Based options; "OJMS" referred to JMS interface to Streams Advanced Queuing (AQ).

For this release, the "OracleAS JMS" and "OJMS" nomenclature is not used. The "Oracle Enterprise Messaging Service (OEMS) JMS" reference is used instead. This change reflects the fact that Oracle offers a single Java Messaging Service (JMS) interface to the three message persistence options. As a result, you do not have to change your JMS application code if you decide to change message persistence between any of the three quality of service choices.

3.5.2 Using the JMS Connector Provided by 10*g* Release 3 (10.1.3)

Oracle Application Server 10*g* Release 3 (10.1.3) provides a J2CA 1.5-compliant resource adapter called the JMS Connector that allows OC4J-managed applications to have a unified mechanism to access any JMS provider that implements JMS 1.1 or 1.2b.

Out-of-the-box, this release provides `OracleASjms`, which is an instance of the JMS Connector that is pre-configured for use with the OEMS JMS In-Memory and File-Based options.

Before you redeploy 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) J2EE applications that use JMS in global transactions, you must modify the corresponding deployment descriptors to use OEMS JMS In-Memory and File-Based options via the JMS Connector. For 10*g* Release 3 (10.1.3), OEMS JMS In-Memory and File-Based options cannot be used for global transactions without the JMS Connector.

Oracle recommends that new JMS applications be deployed using the JMS Connector. The JMS Connector provides the new features introduced in 10*g* Release 3 (10.1.3). Oracle will continue to support JMS applications deployed using the older proprietary OC4J Resource Provider supported in Oracle Application Server 10*g* (9.0.4) and 10*g* Release 2 (10.1.2), but you are strongly encouraged to use the JMS Connector.

> **See Also:** "JMS Connector" in the "Java Message Service (JMS)" chapter of the *Oracle Containers for J2EE Services Guide*

3.5.3 Using the Application Server Control Console to Configure OEMS JMS

Unlike previous versions of Oracle Application Server, you can use the 10*g* Release 3 (10.1.3) Application Server Control Console to manage the OC4J-provided OEMS In-Memory and File-Based resource provider. For example, you can use Application Server Control to create connection factories and destinations, as well as modify specific OEMS configuration properties.

Note that in the Application Server Control Console, the OEMS JMS In-Memory and File-Based resource provider is still referred to as the OracleAS JMS provider.

> **See Also:** "Managing the OracleAS JMS Provider" in the Application Server Control online help

3.5.4 Changes to the jms.xml Configuration File

Oracle Application Server 10*g* Release 3 (10.1.3) introduces additional elements to the `jms.xml` configuration file, as well as changes to the format of the `jms.xml` file so it is compliant with the latest schema.

If you redeploy a JMS application on 10*g* Release 3 (10.1.3), Oracle Application Server automatically rewrites the `jms.xml` file to use the new configuration file format and to add additional queues if they do not exist already.

Specifically, Oracle Application Server adds additional queues that are required by the scheduler and router, and one queue that is added for demonstration purposes. The new queues defined in the updated `jms.xml` file include:

- `jms/RAExceptionQueue`
- `jms/events`
- `jms/jobstore`
- `jms/notifications`

> **See Also:** "Configuration Elements" in the *Oracle Containers for J2EE Services Guide*

3.5.5 List of JAR Files Required for OEMS JMS Lookup

When you redeploy JMS applications on Oracle Application Server 10*g* Release 3 (10.1.3), note the following.

When using OEMS JMS In-Memory and File-Based options directly from an application client, the JAR files that must be included in the class path are listed in Table 3–5, "Client-side JAR Files Required for OEMS JMS In-Memory and File-Based Lookup" in the *Oracle Containers for J2EE Services Guide*.

When using OEMS JMS Database option directly from an application client, the JAR files that must be included in the class path are listed in Table 3-7, "Client-side JAR Files Required for OEMS JMS Database Lookup" in the *Oracle Containers for J2EE Services Guide*.

3.5.6 Database Version Support for OEMS JMS Database

Refer to the "OEMS JMS Database Certification Matrix" in the *Oracle Containers for J2EE Services Guide* for information on which versions of the Oracle database work with the Oracle Application Server when the OJMS client is running in OC4J.

3.6 Java Transaction API (JTA) Considerations

The Java Transaction API (JTA) is a specification developed by Sun Microsystems to provide support for global (distributed) transactions in the J2EE environment. Global transactions combine multiple enterprise systems - such as databases and message queues - into a single unit of work. The JTA maps the specifications based on the Open Group Distributed Transaction Processing model into the Java environment.

> **See Also:** "OC4J Transaction Support" in the *Oracle Containers for J2EE Services Guide*

The following sections highlight key changes to the OC4J JTA Support for 10*g* Release 3 (10.1.3). You should review these sections before deploying your existing J2EE application on 10*g* Release 3 (10.1.3):

3.6.1 Using the New Middle-Tier Two-Phase Commit (2PC) Coordinator Instead of the Database Transaction Coordinator

Oracle Application Server 10*g* Release 3 (10.1.3) introduces the Middle-Tier Two-Phase Commit (2PC) Coordinator that supports all XA-compatible resources, not just those from Oracle. This feature is referred to as a "heterogeneous middle tier coordinator".

As a result, you are encouraged to use this new 2PC coordinator, instead of the deprecated in-database two-phase commit coordinator.

> **See Also:** "Middle-Tier Two-Phase Commit (2PC) Coordinator" in the *Oracle Containers for J2EE Services Guide*

3.6.2 New Support for Transaction Propagation

OC4J 10*g* Release 3 (10.1.3) introduces JTA transaction propagation. Transaction context propagation makes it possible for multiple OC4J instances to participate in a single global transaction.

Previous versions of Oracle Application Server did not support transaction propagation. As a result, when an OC4J instance that supports transaction propagation makes a remote method invocation on a bean that is deployed on an older version of OC4J that does not support transaction propagation, no transaction context is propagated.

> **See Also:** "Transaction Propagation Between OC4J Processes Over ORMI" in the *Oracle Containers for J2EE Services Guide*

3.7 Remote Method Invocation (RMI) Considerations

Oracle Application Server 10*g* Release 3 (10.1.3) supports several new features and changes to the OC4J Remote Method Invocation (RMI) implementation. For more information, see the following sections:

3.7.1 Applying Compatibility Patches for 10*g* (9.0.4) and 10*g* Release 2 (10.1.2)

To use ORMI to invoke a method on a remote object when the invoking object and the invoked object are running on different OC4J versions, you must install a patch on the older version. This applies when the newer version is 10*g* Release 3 (10.1.3) and the olde version is 10*g* (9.0.4) or 10*g* Release 2 (10.1.2).

For more information, see "Compatibility Patches for 9.0.4.x and 10.1.2.x" in the *Oracle Containers for J2EE Services Guide*.

3.7.2 New System Property for Configuring ORMI Request Load Balancing

In previous releases, when two or more clients in the same process retrieved an `InitialContext`, you could use the `dedicated.connection` or `dedicated.rmicontext` properties to be sure that each client received its own `InitialContext` instead of a shared context. When each client had its own `InitialContext`, then the clients could be load balanced.

These properties are deprecated in 10*g* Release 3 (10.1.3). Instead, you should use the new `oracle.j2ee.rmi.loadBalance` system property to specify load balancing in an application cluster. This property can be set in the client's `jndi.properties` file or in a Hashtable in the client code. The values for this property are:

- `client` — The client interacts with the OC4J process that was initially chosen at the first lookup (this is the default setting).
- `context` — The client goes to a new server when a separate context is used (this is similar to the deprecated `dedicated.rmicontext` property).
- `lookup` — The client goes to a new (randomly selected) server for every request.

3.7.3 New Implementation of ORMI Tunnelling through HTTP

Oracle Application Server 10*g* Release 3 (10.1.3) introduces a new implementation for ORMI tunneling through HTTP. For complete information, see "Configuring ORMI Tunneling through HTTP" in the *Oracle Containers for J2EE Services Guide*.

3.7.4 Configuring Secure Connections with RMIS and SSL

Oracle Application Server 10*g* Release 3 (10.1.3) supports the use of Secure Socket Layer (SSL) for RMI connections. Complete instructions for configuring RMIS for your OC4J instances is included in the *Oracle Containers for J2EE Security Guide*.

Besides securing the RMI connections for your deployed applications, you can also secure the RMI management connections between the Administration OC4J instance (which is used to deploy the Application Server Control Console) and the other OC4J instances you are managing. For more information, see "Configuring Security for the Application Server Control Console" in the *Oracle Application Server Administrator's Guide*.

3.8 Java Naming and Directory Interface (JNDI) Considerations

Oracle Application Server 10*g* Release 3 (10.1.3) introduces several new features and changes to JNDI for this release. For a complete list of the new and changed JNDI features, see "Oracle JNDI" in the *Oracle Containers for J2EE Services Guide*.

In particular, before you deploy your J2EE applications on 10*g* Release 3 (10.1.3), review the following sections:

- New Package Names for Initial JNDI Context Factories
- JNDI-Related MBeans Now Available in the Application Server Control Console
- Performing Inter-Application JNDI Lookups
- Browsing the JNDI Context in the Application Server Control Console

3.8.1 New Package Names for Initial JNDI Context Factories

Oracle Application Server 10*g* (9.0.4) and 10*g* Release 2 (10.1.2) package names for OC4J initial context factories are deprecated. They will no longer be supported in future releases. Specifically, the following context factories are deprecated:

```
com.evermind.server.rmi.RMIInitialContextFactory
com.evermind.server.ApplicationClientInitialContextFactory
com.oracle.iiop.server.IIOPInitialContextFactory
```

Instead, you should use the following settings when using the `java.naming.factor.initial` property:

```
oracle.j2ee.rmi.RMIInitialContextFactory
oracle.j2ee.naming.ApplicationClientInitialContextFactory
oracle.j2ee.iiop.IIOPInitialContextFactory
```

> **See Also:** "Initial Context" in the *Oracle Containers for J2EE Services Guide*

3.8.2 JNDI-Related MBeans Now Available in the Application Server Control Console

The following JNDI-related MBeans are now registered with OC4J and are available for use within the MBean browser in the Application Server Control Console:

- JNDIResource
- JNDINamespace

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

3.8.3 Performing Inter-Application JNDI Lookups

It is now possible to configure JNDI to perform inter-application lookups. This in contrast to the default behavior, where lookups within an application are bound to be available within the current application's namespace.

Note that for global lookup to work properly, the target application's classes must be in the classpath of the application attempting the lookup.

> **See Also:** "Configuring JNDI for Deployment" in the *Oracle Containers for J2EE Services Guide*

3.8.4 Browsing the JNDI Context in the Application Server Control Console

You can now browse the JNDI context for a selected application with the 10*g* Release 3 (10.1.3) Application Server Control Console.

To browse the JNDI context, select the **JNDI Browser** task on the OC4J Administration page in the Application Server Control Console.

> **See Also:** "Browsing the JNDI Namespace for an OC4J Instance" in the Application Server Control online help

3.9 Security Considerations

Review the following sections for information on providing security for the J2EE applications you deploy on 10*g* Release 3 (10.1.3):

- List of Significant Changes in OC4J Security for 10g Release 3 (10.1.3)

- Converting principals.xml to the New JAAS Security Model
- Using Oracle Internet Directory as a Security Provider

3.9.1 List of Significant Changes in OC4J Security for 10*g* Release 3 (10.1.3)

Before you redeploy your applications on 10*g* Release 3 (10.1.3), consider the following changes in the OC4J security features for this release:

- This release of OracleAS JAAS Provider requires JDK1.4.
- There is a new consolidated "JAAS mode" for authorization, for both servlets and EJBs. This replaces previous `runas-mode` and `dosasprivileged-mode` functionality for servlets, and `USE_JAAS` functionality (introduced in preliminary 10.1.3 releases) for EJBs. The previous functionality is supported but deprecated in the OC4J 10*g* Release 3 (10.1.3) implementation.

  See Also: "JAAS Authorization and JAAS Mode" in the *Oracle Containers for J2EE Security Guide*

- The `jazn-data.xml` configuration file used in previous releases to store user and role configuration (for the file-based provider), policy configuration (for the file-based, external LDAP, or custom security provider), and login module configuration (for all security providers) has been renamed `system-jazn-data.xml`. However, an application can optionally use an application-specific `jazn-data.xml` repository file to store user and role configuration for the file-based provider.
- The `XMLUserManager` class and its datastore, `principals.xml`, are still supported for this release, but you are strongly advised to migrate to the new JAAS Security model. The `principals.xml` file will no longer be supported in future releases.

  See Also: "Converting principals.xml to the New JAAS Security Model"

- Most of the classes in the `com.evermind` package have been replaced by `oracle.j2ee`. Although the `com.evermind.*` classes continue to exist, they are deprecated; Oracle encourages you to move your applications to `oracle.j2ee.*` as soon as possible.

  The two exceptions are the following classes in the com.evermind package, which will be moved to the `oracle.j2ee` package in a future release:

  ```
  com.evermind.server.rmi.RMIPermission
  com.evermind.server.AdministrationPermission
  ```

- Custom `UserManager` classes are supported in this release, but will be unsupported in future releases.
- The application realm and external realm are deprecated.

  See Also: "Standard Security Concepts" in the *Oracle Containers for J2EE Security Guide*

3.9.2 Converting principals.xml to the New JAAS Security Model

For Oracle Application Server 10*g* Release 3 (10.1.3), the `XMLUserManager` class and its datastore, `principals.xml`, are supported in this release, but you are strongly encouraged to migrate to the new JAAS security model.

If an application that you want to redeploy on 10*g* Release 3 (10.1.3) was previously using the `XMLUserManager` class, you can use the JAZN Admintool to migrate the data in the principals defined in the `principals.xml` file to the new JAAS security model.

For more information, see "Migrating Principals from the principals.xml File" in the *Oracle Containers for J2EE Security Guide*.

3.9.3 Using Oracle Internet Directory as a Security Provider

Oracle Application Server 10*g* Release 3 (10.1.3) supports the use of Oracle Internet Directory as a security provider and OracleAS Single Sign-On for the applications you deploy.

Before you deploy an application that requires Oracle Internet Directory or OracleAS Single Sign-On, see "Oracle Identity Management Security Provider" in the *Oracle Containers for J2EE Security Guide* for complete instructions.

> **See Also:** Section 4.2, "Using Existing OracleAS Infrastructure Components"

3.10 Oracle TopLink and EJB Considerations

Use the following sections to take advantage of Oracle TopLink for your 10*g* Release 3 (10.1.3) applications:

- Configuring CMP Entity Beans to Use Oracle TopLink Persistence Manager
- Upgrading TopLink Workbench Projects

3.10.1 Configuring CMP Entity Beans to Use Oracle TopLink Persistence Manager

In previous releases, the default persistence manager was Orion CMP. In 10g Release 3 (10.1.3), OC4J is configured by default to use Oracle TopLink as its default persistence manager.

As a result, before you redeploy your EJB applications that use Orion CMP, you must migrate persistence configuration from your original `orion-ejb-jar.xml` file to the `toplink-ejb-jar.xml` file.

Oracle provides a TopLink migration tool that you can use to automate this migration.

> **See Also:** "Migrating OC4J Orion Persistence to OC4J TopLink Persistence" in the *Oracle TopLink Developer's Guide*

3.10.2 Upgrading TopLink Workbench Projects

If you have used Oracle TopLink with previous versions of Oracle Application Server, you can migrate your existing Oracle TopLink projects to TopLink 10g Release 3 (10.1.3).

For more information, see "Migrating to 10g Release 3 (10.1.3)" in the *Oracle TopLink Getting Started Guide*.

4

Integrating 10*g* Release 3 (10.1.3) into Your Existing Environment

If you are already using Oracle Application Server 10*g* (9.0.4) or 10*g* Release 2 (10.1.2), use this chapter to understand how to integrate Oracle Application Server 10*g* Release 3 (10.1.3) into your existing Oracle Application Server environment.

The following sections describe how you can use your existing Oracle Application Server components with your new 10*g* Release 3 (10.1.3) installations:

- Installing 10g Release 3 (10.1.3) Middle Tiers in an Existing Oracle Application Server Environment
- Using Existing OracleAS Infrastructure Components
- Using Existing Oracle HTTP Server Instances
- Using an Existing OracleAS Web Cache Installation As a Reverse Proxy

4.1 Installing 10*g* Release 3 (10.1.3) Middle Tiers in an Existing Oracle Application Server Environment

There are no restrictions to prevent you from installing 10*g* Release 3 (10.1.3) middle tiers in an existing 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) environment. In fact, if necessary, you can install your new 10*g* Release 3 (10.1.3) middle tiers on the same host as your existing 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) Oracle homes.

As with other Oracle software products, Oracle Universal Installer will check for available ports and perform additional prerequisite checks to be sure the host computer meets the 10*g* Release 3 (10.1.3) hardware and software requirements.

However, while 10*g* Release 3 (10.1.3) middle tiers can co-exist with previous Oracle Application Server releases, there are fundamental differences between these releases. For example, 10*g* Release 3 (10.1.3) introduces significant changes to clustering application servers. For more information refer to the following:

- Chapter 5, "Differences Between 10g Release 3 (10.1.3) and Previous Releases"
- Chapter 6, "10g Release 3 (10.1.3) Version Compatibility"

4.2 Using Existing OracleAS Infrastructure Components

Your existing 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) OracleAS Infrastructure consists of an OracleAS Metadata Repository and OracleAS Identity Management.

Oracle Application Server 10*g* Release 3 (10.1.3) does not require an OracleAS Metadata Repository because none of the components delivered with this release require specific schemas or a database to host those schemas.

However, there are many reasons to configure your 10*g* Release 3 (10.1.3) middle tier to use your existing OracleAS Identity Management installation. Table 4–1 lists the tasks you can accomplish when you use OracleAS Identity Management with 10*g* Release 3 (10.1.3).

To configure your 10*g* Release 3 (10.1.3) instance to use OracleAS Identity Management, select the **Identity Management** task on the OC4J Administration page in the Application Server Control Console.

> **See Also:** "Configuring Instances to Use 10.1.2 Oracle Identity Management" in the *Oracle Application Server Administrator's Guide*

Table 4–1 Using OracleAS Identity Management with 10g Release 3 (10.1.3)

Task	For More Information
Use Oracle Internet Directory as the security provider for the J2EE applications you deploy.	"Overview of Managing Security Providers" in the Application Server Control online help
Use Oracle Internet Directory as the security provider for the 10*g* Release 3 (10.1.3) Application Server Control. You can then use Oracle Internet Directory to manage the administrator accounts that are used to log in to the Application Server Control Console.	"Configuring the Security Provider for Application Server Control" in the Application Server Control online help
Configure your J2EE applications to use OracleAS Single Sign-On	"About Using OracleAS Single Sign-On" in the Application Server Control online help
Use 10*g* Release 2 (10.1.2) Single Sign-on to authenticate connections between a Web application and its Web services. In this scenario, it is assumed that both the Web application and the Web service are configured to use the Security Assertion Markup Language (SAML).	"Configuring Single Sign-on Using SAML" in the *Oracle Application Server Web Services Security Guide*

4.3 Using Existing Oracle HTTP Server Instances

When you install Oracle Application Server 10*g* Release 3 (10.1.3), you can choose to install Oracle HTTP Server in one or more of your 10*g* Release 3 (10.1.3) Oracle homes. You can then configure your environment to use the Oracle HTTP Server to serve as a front-end for the J2EE applications you deploy.

Alternatively, you can configure your existing 10*g* Release 2 (10.1.2) Oracle HTTP Server as the front-end to your new 10*g* Release 3 (10.1.3) instances.

For more information, see "Configuring Oracle Application Server 10.1.2 with Oracle Application Server 10.1.3" in the *Oracle Application Server Administrator's Guide*.

4.4 Using an Existing OracleAS Web Cache Installation As a Reverse Proxy

You can use Release 2 (10.1.2) OracleAS Web Cache as a reverse proxy for your 10g Release 3 (10.1.3) middle-tier instance. As a reverse proxy server, OracleAS Web Cache acts a gateway to the middle-tier servers.

For more information, see "Configuring 10.1.2 OracleAS Web Cache as a Reverse Proxy" in the *Oracle Application Server Administrator's Guide*.

5

Differences Between 10*g* Release 3 (10.1.3) and Previous Releases

Use this chapter to learn about the key differences between Oracle Application Server 10*g* Release 3 (10.1.3) and the previous Oracle Application Server releases:

5.1 Using the Cluster Topology Instead of an OracleAS Farm

With previous versions of Oracle Application Server, you can configure a set of Oracle Application Server instances so they use a common OracleAS Metadata Repository. The instances that share the common OracleAS Metadata Repository are members of the same OracleAS Farm. From the OracleAS Farm page in the 10*g* (9.0.4) or 10*g* Release 2 (10.1.2) Application Server Control Console, you can view all the application servers that are members of the OracleAS Farm. In addition, the Distributed Configuration Management (DCM) software provides the underlying technology for managing the OracleAS Farm.

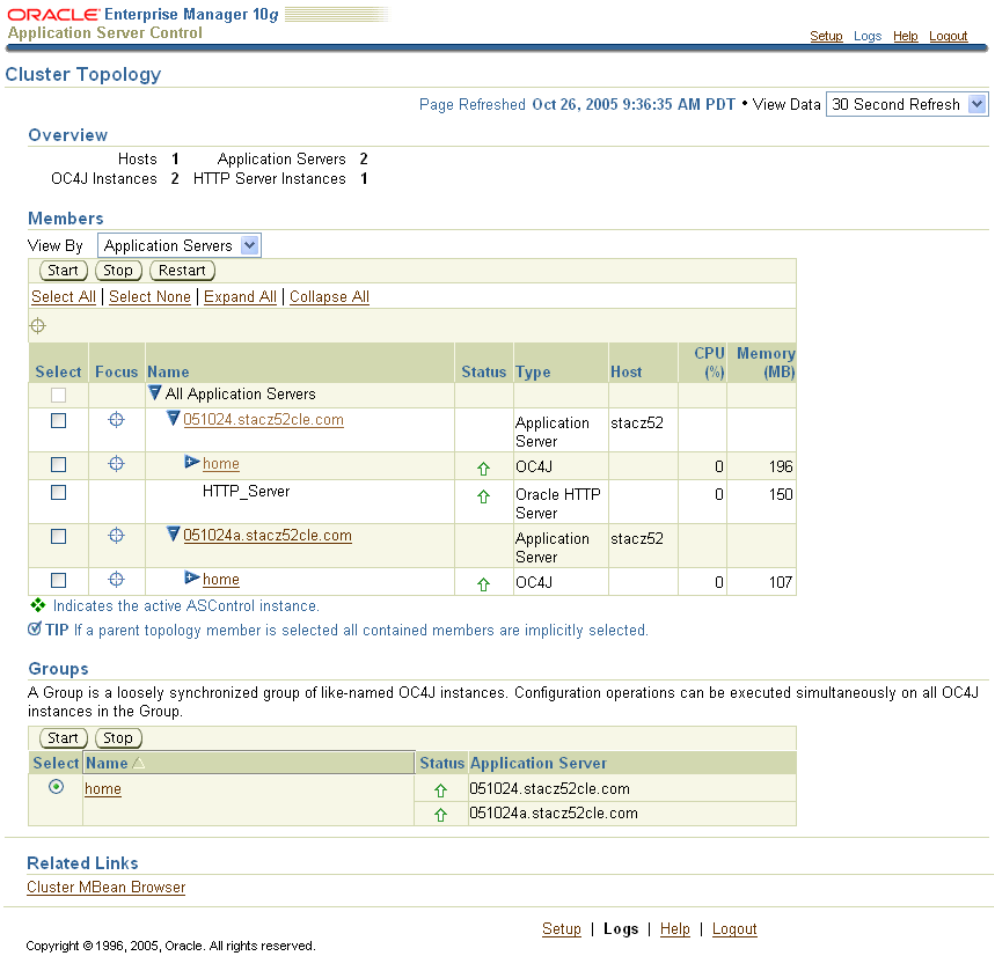
Oracle Application Server 10*g* Release 3 (10.1.3) does not require an OracleAS Metadata Repository or the DCM software. As a result, there is no concept of an OracleAS Farm. Instead, in 10*g* Release 3 (10.1.3), you configure your 10*g* Release 3 (10.1.3) instances so they can communicate via Oracle Process Manager and Notification Server (OPMN).

When you configure two or more 10*g* Release 3 (10.1.3) instances in this manner, the instances can be managed from the Cluster Topology page in the 10*g* Release 3 (10.1.3) Application Server Control Console.

> **See Also:** "Configuring and Managing Clusters" in the *Oracle Containers for J2EE Configuration and Administration Guide*

Figure 5–1 shows the 10*g* Release 3 (10.1.3) Cluster Topology page, which includes two Oracle Application Server instances that have been configured to communicate via the same multicast address and port.

Figure 5–1 Oracle Application Server 10g Release 3 (10.1.3) Cluster Topology Page



5.2 Using Groups Instead of OracleAS Clusters

With previous versions of Oracle Application Server, you create and manage OracleAS Clusters. OracleAS Clusters consist of identically configured J2EE and Web Cache installations that are part of the same OracleAS Farm. Distributed Configuration Management (DCM) is then used to keep the instances within the cluster in synch. Configuration changes made to one instance in the cluster are automatically applied to other instances in the cluster.

In 10*g* Release 3 (10.1.3), there is no OracleAS Farm and there is no DCM. However, you can still group multiple Oracle Containers for J2EE (OC4J) instances that are part of the same cluster topology. These groups of OC4J instances can be used in a similar manner to OracleAS Clusters.

Refer to the following sections for more information:

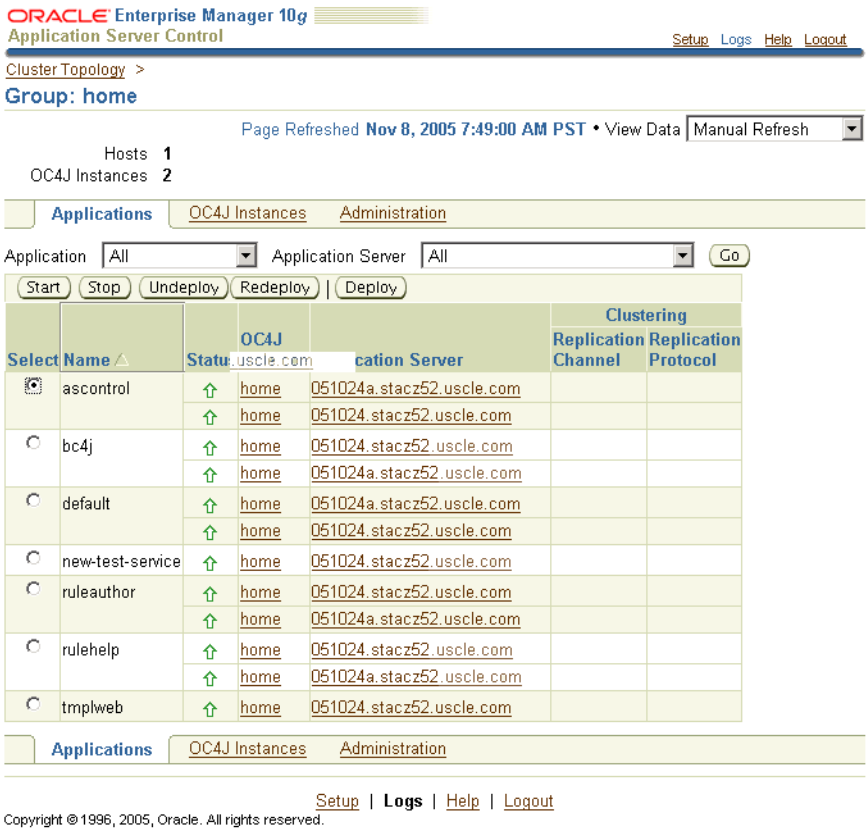
- How Are Groups Similar to OracleAS Clusters?
- How Are Groups Different from OracleAS Clusters?

5.2.1 How Are Groups Similar to OracleAS Clusters?

Like OracleAS Clusters, groups make it easy to deploy your applications to more than one OC4J instance at a time:

- With OracleAS Clusters, changes made to one instance in a cluster are automatically propogated to other instances in the cluster. For example, if you deploy an application to one instance in the cluster, the application is automatically deployed to the other instances.
- With groups, you deploy your J2EE applications to all the OC4J instances in the group using the Group page (Figure 5–2). The Group page is available from the Cluster Topology page.

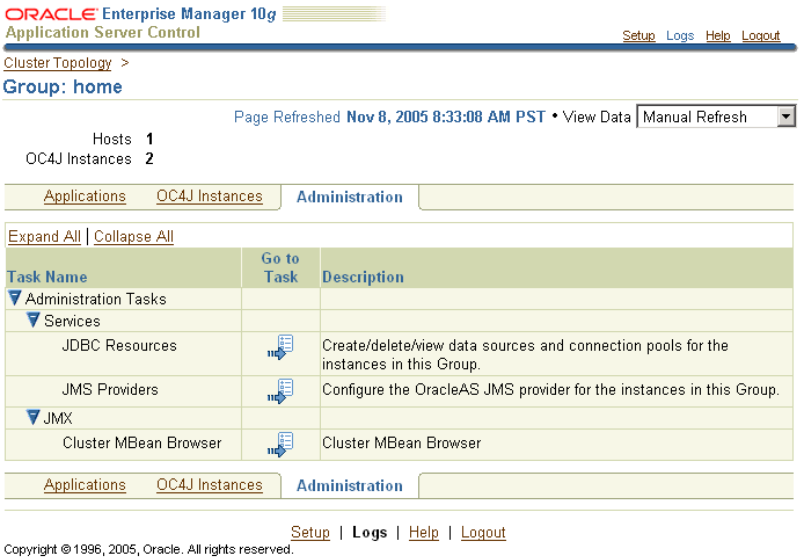
Figure 5–2 Oracle Application Server 10g Release 3 (10.1.3) Group Page



Like OracleAS Clusters, groups allow you to make specific configuration changes to multiple OC4J instances. Specifically, you can use the Group Administration page to configure JDBC data sources and JMS destinations and connection factories for all the OC4J instances in the group, and to access the cluster MBean browser.

Figure 5–3 shows the Group Administration page, which you can use to manage JDBC and JMS configurations across all instances in a group.

Figure 5–3 Oracle Application Server 10g Release 3 (10.1.3) Group Administration Page



5.2.2 How Are Groups Different from OracleAS Clusters?

The following sections describe some key differences between 10*g* Release 3 (10.1.3) groups and 10*g* Release 2 (10.1.2) OracleAS Clusters:

- Configuration Changes Are Not Propogated Automatically to Other Members of the Group
- Groups Are Not Always Identical
- Groups Are Created Automatically Based on the Instance Name

5.2.2.1 Configuration Changes Are Not Propogated Automatically to Other Members of the Group

Unlike OracleAS Clusters, configuration changes made to an individual OC4J instance in a group (from the OC4J instance Home page or from the command line) are not automatically applied to other OC4J instances in the cluster.

Instead, if you want to make a configuration change to all the OC4J instances in a group, you must either make the change from the Group page, or you must use the Application Server Control Console or command line tools to make the change to each OC4J instance in the cluster.

Similarly, if you add a new OC4J instance to the group, configuration changes are not automatically applied to the new instance. Instead, you must use the OC4J home page for that instance, or the command-line tools, to apply any required configuration changes to the new instance.

> **See Also:** "Replicating Changes Across a Cluster" in the *Oracle Containers for J2EE Configuration and Administration Guide*

5.2.2.2 Groups Are Not Always Identical

In some ways, groups provide more flexibility than OracleAS Clusters. For example, when you add an OC4J instance to an cluster in 10*g* (9.0.4) or 10*g* Release 2 (10.1.2), the instance can be used for cluster operations only. Any changes you make to the instance are automatically applied to the other instances in the cluster.

In 10*g* Release 3 (10.1.3), you have the flexibility to deploy an application to just one OC4J instance in the group, or to adjust the attributes of one instance without impacting the other instances in the group.

Note, however, that if you make changes to one member of the group and not to another, some operations performed on the group could succeed on one instance and fail on another.

Also, unlike OracleAS Clusters, actions you perform on a group do not affect OC4J instances within the group that are not up and running when the operation is performed.

5.2.2.3 Groups Are Created Automatically Based on the Instance Name

And finally, the way you create 10*g* Release 3 (10.1.3) groups is very different from the method used to create OracleAS Clusters. You cannot create a new group from the Application Server Control Console or from the command line. Instead, a group is created automatically when two OC4J instances have the same name.

For example, suppose you create an OC4J instance called OC4J1 on one host and create an instance called OC4J1 on another host. If the two application servers are clustered, a new group called OC4J1 is automatically formed. You can display the OC4J1 Group Home page from the Cluster Topology Page in the Application Server Control Console.

5.3 Using Application Clustering for State Replication

In addition to clusters and groups, Oracle Application Server 10*g* Release 3 (10.1.3) introduces the concept of application clustering, which provides state replication and load balancing for applications within your cluster topology.

The following sections provide more information:

- Clustering Features and Concepts That Are No Longer Supported
- About 10g Release 3 (10.1.3) Application Clustering

5.3.1 Clustering Features and Concepts That Are No Longer Supported

Application Clustering provides a simpler, more efficient method of replicating application state, which replaces the following concepts and features that are no longer supported in 10*g* Release 3 (10.1.3):

- Islands

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

- loadbalancer.jar

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

- Deprecated Clustering-Specific XML Elements

The following XML elements are deprecated in OC4J 10g (10.1.3) and should no longer be used to configure clustering. The new `<cluster>` element is now used for all cluster management:

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

5.3.2 About 10*g* Release 3 (10.1.3) Application Clustering

Within a 10*g* Release 3 (10.1.3) cluster, you can configure clustering for selected applications that are deployed across the cluster. Application clustering offers the following features:

- You can configure clustering for specific applications, or globally by configuring clustering for the `default` application in an OC4J instance.

  Other applications deployed to the instance automatically inherit the clustering characteristics of the `default` application.
- You can configure clustering for an application at deployment time, or later, after you deploy the application.
- You can select from the following replication methods:
  - Peer-to-peer replication
  - Multicast replication
  - Database replication

See Also: "Application Clustering in OC4J" in the *Oracle Containers for J2EE Configuration and Administration Guide* for more detailed information about the supported replication methods

5.4 Creating New OC4J Instances With the createinstance Command

Each time you install Oracle Application Server 10*g* Release 3 (10.1.3), the installation procedure automatically installs and configures a single OC4J instance. By default, this instance is called the `home` instance. However, if you use select **Advanced Installation Mode** during the installation, you can provide a custom name for this default OC4J instance.

In previous versions of Oracle Application Server, you could use the Application Server Control Console to create additional OC4J instances. However, in 10*g* Release 3 (10.1.3), you use the new `createinstance` command, which is run from the command-line. The createinstance command is available in the `bin` directory of the 10*g* Release 3 (10.1.3) Oracle home.

For complete instructions on using the `createinstance` command, as well as the related `removeinstance` command, see "Creating and Managing Additional OC4J Instances" in the *Oracle Containers for J2EE Configuration and Administration Guide*.

5.5 Using the admin_client.jar Utility to Manage OC4J Instances and Clusters

OC4J 10*g* Release 3 (10.1.3) also provides a command-line utility— `admin_client.jar`—that can be used to perform operations on active OC4J instances.

For many functions, the `admin_client.jar` utility replaces the `admin.jar` utility, which is used exclusively for the standalone configuration of 10*g* Release 3 (10.1.3) OC4J server.

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

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

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

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

5.6 Summary of Equivalent Features in 10*g* Release 3 (10.1.3)

Table 5–1 describes how some common Oracle Application Server management tasks were performed in prior releases of Oracle Application Server and the equivalent procedures in 10*g* Release 3 (10.1.3).

Table 5–1 Summary of Changed Features for 10g Release 3 (10.1.3)

Task or Feature	In Prior Releases...	In 10*g* Release 3 (10.1.3)...
Clustering Oracle Application Server instances	Configure multiple Oracle Application Server instances so they use the same OracleAS Metadata Repository. This creates an OracleAS Farm, which can be viewed from the Application Server Control Console Farm page.	Run the `opmnassociate` command in each Oracle home, or perform the equivalent task as part of the installation. This causes the selected Oracle Application Server instances to appear on the Cluster Topology page of the Application Server Control Console.
Performing management tasks simultaneously on multiple OC4J instances	Add selected J2EE and Web Cache instances within an OracleAS Farm to an OracleAS Cluster. Perform this task from the Farm page in the Application Server Control Console.	Create multiple OC4J instances that use the same name and reside within the same Cluster Topology page. The like-named OC4J instances appear as Groups on the Cluster Topology page in the Application Server Control Console.
Replicating application state across a cluster	OC4J processes and islands within OracleAS Clusters.	Application clustering, which can be configured from the Application Server Control Console during deployment or post-deployment.

Table 5–1 (Cont.) Summary of Changed Features for 10g Release 3 (10.1.3)

Task or Feature	In Prior Releases...	In 10*g* Release 3 (10.1.3)...
Creating new OC4J instances	Click **Create Instance** on the OC4J Home page in the Application Server Control Console.	Use the `createinstance` command in the `bin` directory of the Oracle Application Server Oracle home.
Using command-line tools to manage instances and clusters	Use one of the following: ■ Distributed Configuration Management (DCM) command line (dcmctl) ■ Oracle Process Manager and Notification Server (OPMN) command line (opmnctl)	DCM is not available in 10*g* Release 3 (10.1.3), but new `opmnctl` commands and the new `admin_client.jar` utility provide additional management capabilities.
Using OracleAS Identity Management	Configure OracleAS Identity Management using the Application Server Infrastructure page in the Application Server Control Console.	Configure OracleAS Identity Management using the Identity Management task on the OC4J Administration page in the Application Server Control Console.

6

10*g* Release 3 (10.1.3) Version Compatibility

Oracle Application Server 10*g* Release 3 (10.1.3) is designed to be installed in parallel to your existing Oracle Application Server installations. For example, you can install a new 10*g* Release 3 (10.1.3) Oracle home on a host where you have already installed Oracle Application Server 10*g* (9.0.4) or Oracle Application Server 10*g* Release 2 (10.1.2).

Oracle Application Server 10*g* Release 3 (10.1.3) does not require an OracleAS Metadata Repository, but you can configure your 10*g* Release 3 (10.1.3) installations to take advantage of an existing 10*g* Release 2 (10.1.2) OracleAS Identity Management installation.

Table 6–1 provides an overview of Oracle Application Server 10*g* Release 3 (10.1.3) compatibility with previous Oracle Application Server releases.

Table 6–1 10g Release 3 (10.1.3) Compatibility With Previous Versions

Component and Release	10*g* Release 3 (10.1.3) Compatibility Information
10*g* (9.0.4) Middle Tiers	■ You can install and run 10*g* (9.0.4) and 10*g* Release 3 (10.1.3) middle tiers on the same host. ■ The two Oracle homes can also share the same OracleAS Identity Management. ■ However, 10*g* (9.0.4) and 10*g* Release 3 (10.1.3) middle-tier instances cannot be in the same cluster. ■ Before you can use ORMI connections between the middle tiers, you must apply a patch on the 10*g* (9.0.4) middle tier. For more information, see Section 3.7.1, "Applying Compatibility Patches for 10g (9.0.4) and 10g Release 2 (10.1.2)"
10*g* (9.0.4.0.x) OracleAS Identity Management	Not supported; you must apply the 10*g* (9.0.4.2) patchset before using the 10*g* (9.0.4) OracleAS Identity Management.
10*g* (9.0.4.1) OracleAS Identity Management	Not supported; you must apply the 10*g* (9.0.4.2) patchset before using the 10*g* (9.0.4) OracleAS Identity Management.
10*g* (9.0.4.2) OracleAS Identity Management	You can configure your 10*g* Release 3 (10.1.3) middle tiers to use an existing 10*g* (9.0.4.2) OracleAS Identity Management only if you have applied the patch for bug 4217661. To obtain the patch, contact Oracle Support. You can access information about technical support at: `http://www.oracle.com/support` See "Configuring Instances to Use 9.0.4 or 10.1.2 Oracle Identity Management" in the *Oracle Application Server Administrator's Guide*.

Table 6–1 (Cont.) 10g Release 3 (10.1.3) Compatibility With Previous Versions

Component and Release	10*g* Release 3 (10.1.3) Compatibility Information
10*g* Release 2 (10.1.2) Middle Tiers	■ You can run 10*g* Release 2 (10.1.2) and 10*g* Release 3 (10.1.3) middle tiers on the same host. ■ The two Oracle homes can also share the same OracleAS Identity Management. ■ However, 10*g* Release 2 (10.1.2) and 10*g* Release 3 (10.1.3) middle-tier instances cannot be in the same cluster. ■ Before you can use ORMI connections between the middle tiers, you must apply a patch on the 10*g* (9.0.4) middle tier. For more information, see Section 3.7.1, "Applying Compatibility Patches for 10g (9.0.4) and 10g Release 2 (10.1.2)"
10*g* Release 2 (10.1.2.0.0) OracleAS Identity Management	Not supported; you must apply the 10*g* Release 2 (10.1.2.1.0) patchset.
10*g* Release 2 (10.1.2.0.1) Standard Edition One OracleAS Identity Management	You can configure your 10*g* Release 3 (10.1.3) middle tiers to use an existing 10*g* Release 2 (10.1.2.0.1) OracleAS Identity Management. See "Configuring Instances to Use 9.0.4 or 10.1.2 Oracle Identity Management" in the *Oracle Application Server Administrator's Guide*.
10*g* Release 2 (10.1.2.0.2) OracleAS Identity Management	You can configure your 10*g* Release 3 (10.1.3) middle tiers to use an existing 10*g* Release 2 (10.1.2.0.2) OracleAS Identity Management. See "Configuring Instances to Use 9.0.4 or 10.1.2 Oracle Identity Management" in the *Oracle Application Server Administrator's Guide*.
10*g* Release 2 (10.1.2.1.0) OracleAS Identity Management	You can configure your 10*g* Release 3 (10.1.3) middle tiers to use an existing 10*g* Release 2 (10.1.2.1.0) OracleAS Identity Management. See "Configuring Instances to Use 9.0.4 or 10.1.2 Oracle Identity Management" in the *Oracle Application Server Administrator's Guide*.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

O

P

R

S

T

U

V

W

X