Oracle® Application Server 10*g*

Forms and Reports Services Upgrade Guide

10*g* (9.0.4) for UNIX

Part No. B13547-01

December 2003

ORACLE®

Oracle Application Server 10*g* Forms and Reports Services Upgrade Guide 10*g* (9.0.4) for UNIX

Part No. B13547-01

Primary Author: Julia Pond

Contributors: Chung Cheng, Orlando Cordero, Bob Donat, Shaun Lin, Jeremy Lizt, Regis Louis, Valarie Moore, Oscar Naim, Bill Norcott, Shane Potter, Pavi Sandhu, Charlie Shapiro, Jianping Shi, Debbie Steiner, Jeff Tang, Parthiban Thilagar, Ravikumar Venkatesan, Richard Wang, Brian Wright, Liujin Yu

Contents

3 Completing the Upgrade

Index

Send Us Your Comments

Oracle Application Server 10*g* Forms and Reports Services Upgrade Guide 10*g* (9.0.4) for UNIX

Part No. B13547-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: formsdoc_us@oracle.com
- Postal service:

  Oracle Corporation
  Oracle Forms Documentation
  200 Oracle Parkway, 2op12
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This guide contains information on upgrading Oracle Forms and Reports applications from Oracle9*i*AS Release 2 (9.0.2) Server 9.0.2 to OracleAS Forms and Reports Services 10*g* (9.0.4).

Intended Audience

This guide is intended for developers, system administrators, and DBAs who develop, deploy, and manage Oracle Application Server Forms Services and Oracle Application Server Reports Services applications in Oracle Application Servers.

Structure

Chapter 1, "Performing an Upgrade"

This chapter describes the OracleAS Forms and Reports Services upgrade process, and explains how to use the OracleAS Upgrade Assistant to perform the majority of the Oracle Application Server Forms Services upgrade.

Chapter 2, "Troubleshooting"

This chapter contains information you need to know about troubleshooting and resolving errors that may occur during an upgrade.

Chapter 3, "Completing the Upgrade"

This chapter describes tasks you may need to perform to complete the upgrade after the OracleAS Upgrade Assistant has finished processing. Some or all of these may be necessary, depending on the configuration upgraded.

Related Documents

For more information, see the following books in the Oracle Application Server documentation set:

- *Oracle Application Server 10g Upgrading to 10g (9.0.4)*
- *Oracle Application Server Forms and Reports Services 10g (9.0.4) Release Notes*
- *Oracle Application Server Forms and Reports Services 10g (9.0.4) Installation Guide*

Conventions

When discussing upgrade, it is necessary to distinguish between the Oracle home of the previous release and the Oracle home of the new release. In this guide, the previous release's Oracle home is called the source Oracle home. The new release's Oracle home is called the destination Oracle home. Typically, it is also necessary to specify the Oracle home as a full path in file locations. The convention for the Oracle home path is included in these typographic conventions used in the guide:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
monospace text	File names, path names, command names, code, URLs
`monospace bold text`	Typed user input.
`monospace italic text`	Variables in text or code.
`<source_MT_OH>`	The full path to the Release 2 (9.0.2) middle tier Oracle home.
`<destination_MT_OH>`	The full path to the 10*g* (9.0.4) middle tier Oracle home.

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. 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 additional information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, 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, JAWS 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.

1 Performing an Upgrade

This chapter describes the Forms and Reports Services upgrade process, and explains how to use the OracleAS Upgrade Assistant to perform the majority of the Oracle Application Server Forms Services upgrade. The following topics are discussed:

1.1 The Forms and Reports Services Upgrade Process

This section provides a complete overview of the Forms and Reports upgrade process, and the prerequisite conditions for upgrading. The processing performed by the OracleAS Upgrade Assistant for each component is also summarized, with references to any manual tasks you may need to perform after the OracleAS Upgrade Assistant has finished processing.

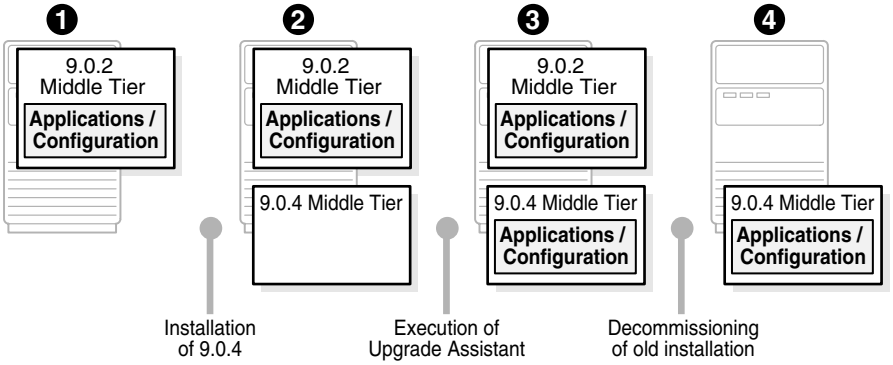
1.1.1 An Overview of the Forms and Reports Upgrade Process

The Forms and Reports Services upgrade process is illustrated in Figure 1–1. The middle tier upgrade consists of the following steps:

1. The Release 2 (9.0.2) Unified Messaging or Business Intelligence and Forms installation resides on its computer, containing applications and configuration data.
2. A 10*g* (9.0.4) middle tier installation of the Forms and Reports Services installation type is installed into a new Oracle home on the same computer.
3. The OracleAS Upgrade Assistant, a tool installed with OracleAS that automates most Forms and Reports Services upgrade tasks, is executed. It copies the applications and configuration data from the source middle tier installation to the 10*g* (9.0.4) installation. Manual tasks may be necessary to complete the upgrade of some configurations.
4. (Optional) The source installation is decommissioned.

Note: The installation of the 10*g* (9.0.4) middle tier will use some of the same port numbers as the Release 2 (9.0.2) middle tier. If you plan to use both middle tiers, you must re-assign port numbers in one of the middle tiers to avoid port conflicts. The components affected are Oracle Enterprise Manager, Oracle HTTP Server, and Oracle Application Server Web Cache.

Figure 1–1 Middle Tier Upgrade



1.1.2 Before You Begin The Upgrade

Before you perform any upgrade tasks, you must ensure that the installations you are working with comprise a supported upgrade path, and that all necessary conditions are in effect in the installations.

Note: When upgrading from an installation of Oracle *i*AS 9.0.2 BI and Forms, and if this installation has Single Sign-On enabled, the OHS configure file 'httpd.conf' will have the following:

```
include
"/private/iasupgd/bifM21j/Apache/Apache/conf/mod_
osso.conf"
```

After upgrading to the Forms and Reports Services install type, this include statement will still exist. Since the Forms and Reports Services install type does not have SSO, you can safely comment out this line in httpd.conf.

1.1.2.1 Supported Upgrade Paths for Forms and Reports Services

Table 1–1 shows the releases, installation types, and topologies of the upgrade paths supported for the Forms and Reports Services upgrade.

Table 1–1 Forms and Reports Services Upgrade Paths

Source Oracle Home	Destination Oracle Home
Oracle9*i*AS Release 2 (9.0.2) Business Intelligence and Forms that uses an Infrastructure	OracleAS 10*g* (9.0.4) Forms and Reports Services
Oracle9*i*AS Release 2 (9.0.2) Business Intelligence and Forms that does not use an Infrastructure	OracleAS 10*g* (9.0.4) Forms and Reports Services
Oracle9*i*AS Release 2 (9.0.2) Unified Messaging that uses an Infrastructure	OracleAS 10*g* (9.0.4) Forms and Reports Services
Oracle9*i*AS Release 2 (9.0.2) Unified Messaging that does not use an Infrastructure	OracleAS 10*g* (9.0.4) Forms and Reports Services

1.1.2.2 Requirements for Upgrade

The OracleAS 10*g* (9.0.4) Forms and Reports Services installation must adhere to the following requirements:

- The Oracle Application Server 10*g* (9.0.4) Forms and Reports Services must be on the same computer as the Oracle9*i*AS Release 2 (9.0.2) product.
- The Oracle Application Server 10*g* (9.0.4) Forms and Reports Services must be installed by the same operating system user that installed the Oracle9*i*AS Release 2 (9.0.2) product.
- The Oracle Application Server 10*g* (9.0.4) Forms and Reports Services must be in a separate Oracle home from the Oracle9*i*AS Release 2 (9.0.2) product.
- Only components that are configured in both the source and the destination Oracle homes will be upgraded.

1.1.2.3 OracleAS Upgrade Assistant Forms and Reports Services Processing

Table 1–2 describes all automated and manual processing performed in the Forms and Reports Services upgrade.

Table 1–2 Components Upgraded by the OracleAS Upgrade Assistant (in processing order) and Possible Manual Upgrade Tasks

Component	The OracleAS Upgrade Assistant...	Manual Tasks
Oracle Process Management and Notification	■ Converts the <*source_MT_OH*>/opmn/conf/opmn.xml file to the 10*g* (9.0.4) format. ■ Merges the converted <*source_MT_OH*>/opmn/conf/opmn.xml file with the <*destination_MT_OH*>/opmn/conf/opmn.xml file. Inserts all custom nodes into <*destination_MT_OH*>/opmn/conf/opmn.xml except the node containing gid="dcm-daemon". ■ **Note:** The OracleAS Upgrade Assistant does not upgrade any changes that were made to Oracle Application Server Containers for J2EE instances in opmn.xml. This includes the instances created by the installer (home, OC4J_WIRELESS, OC4J_DEMOS, OC4J_PORTAL OC4J_BI_FORMS) and instances created by users. The OC4J upgrade process upgrades the home instance and any user-defined instances with applications deployed in the source Oracle home. The other installer-created OC4J instances adopt the Oracle Application Server 10*g* (9.0.4) settings in opmn.xml. If you want to preserve settings from Oracle9*i*AS Release 2 (9.0.2), you must create them manually.	None required.
Oracle Application Server Containers for J2EE	■ The OracleAS Upgrade Assistant creates a separate process, which connects to the source Oracle home. ■ The process uses its Distributed Configuration Management version to examine the instances selected for upgrade. ■ Distributed Configuration Management creates a list of the instances in the source Oracle home in which the applications are deployed. It ignores Oracle-specific OC4J instances, such as OC4J_Portal, defined in the <*source_MT_OH*>/j2ee/deploy.ini* file. The instances are the OC4J upgrade candidates. ■ Distributed Configuration Management builds a list of EAR files for the applications listed. ■ The OracleAS Upgrade Assistant creates a backup of upgraded files, appending a preUpgrade suffix. If necessary, to create a unique file name, it appends an integer, for example, <*file name*>.preUpgrade.1. ■ The OracleAS Upgrade Assistant copies principals.xml, data-sources.xml, jazn-data.xml, and jazn.xml to the destination Oracle home. ■ Adds properties defined in the oc4j.properties file to the opmn.xml file, using the SMI API. ■ The OracleAS Upgrade Assistant rebuilds, and then redeploys the EAR files to the destination Oracle home. In this step, the OracleAS Upgrade Assistant searches for all orion-specific files in the application-deployments directory of the applications. It also searches for application-specific configuration files, such as principals.xml and jazn-data.xml. ■ Distributed Configuration Management updates mod_oc4j.conf with the mount points associated with each deployed application.	Section 3.1, "Completing the Oracle Application Server Containers for J2EE (OC4J) Upgrade" on page 3-1

Table 1–2 Components Upgraded by the OracleAS Upgrade Assistant (in processing order) and Possible Manual Upgrade Tasks

Component	The OracleAS Upgrade Assistant...	Manual Tasks
Oracle HTTP Server	▪ Copies the `httpd.conf` file from the source Oracle home to the destination Oracle home, replacing the *<source_MT_OH>* path with *<destination_MT_OH>*, then applies 10*g* (9.0.4) file changes, and customizations made since Release 2 (9.0.2), to the corresponding file in the destination Oracle home. ▪ Searches the `mod_oc4j.conf` file in *<source_MT_OH>* for `Oc4jMount` directives and copies the `Oc4jMount` directives that contain the string `ajp13://,cluster://` or `instance://` to the `mod_oc4j.conf` file in *<destination_MT_OH>*. ▪ Copies the `mod_osso.conf` file from the source Oracle home to the destination Oracle home, replacing the source Oracle home path with the destination Oracle home path. The `osso.conf` file referenced by the OssoConfigFile directive will be copied and converted into 10*g* (9.0.4) obfuscated files. ▪ Copies the `moddav.conf` file from the source Oracle home to the destination Oracle home, replacing the source Oracle home path with the destination Oracle home path. ▪ Searches `Include` directives in the `httpd.conf` file recursively to locate user-defined configuration files; copies these files from the source Oracle home to `the destination Oracle home`. If the files were found in the source Oracle home, the OracleAS Upgrade Assistant replaces the source Oracle home path with the destination Oracle home path. If the files were found outside of the source `Oracle home`, the OracleAS Upgrade Assistant saves a copy of the original file with a `.preUpgrade` extension in the destination Oracle home, then replaces the source Oracle home file with the new file. ▪ Searches the `LoadModule` directives recursively to find related module dynamic libraries; copies the libraries from the source Oracle home to the destination Oracle home. ▪ Searches the `SSLWallet` directives recursively to find Oracle wallets; copies the wallets from the source Oracle home to the destination Oracle home. ▪ Locates CGI and fastcgi scripts by searching all configuration files for directories and files named in `ScriptAlias` or `ScriptAliasMatch` directives, and the `ExecCGI` option in `Options` directives (defined in `Directory` or `File` containers). Copies the directories and files from the source Oracle home to the destination Oracle home. ▪ Copies static document directories found in the (non-default) location specified by the `DocumentRoot` directive from the source Oracle home to the destination Oracle home. If the `DocumentRoot` directive is the default, static documents are not upgraded. ▪ **Note:** Web sites are often configured with Web Cache as the first listener; in these cases, the Oracle HTTP Server's Listen port may need to be synchronized with equivalent Web Cache port values after upgrade. The settings are shown in Table 3–1, "Oracle HTTP Server and Oracle Application Server Web Cache Port Settings" on page 3-9. The OracleAS Upgrade Assistant does not upgrade the `oracle_apache.conf` and `mod_plsql.conf` files. Static files referenced by the `Alias` or `mod_rewrite` directives are not upgraded. Any such files in an Oracle home must be upgraded manually.	Section 3.2, "Completing the Oracle HTTP Server Upgrade" on page 3-8

Table 1–2 Components Upgraded by the OracleAS Upgrade Assistant (in processing order) and Possible Manual Upgrade Tasks

Component	The OracleAS Upgrade Assistant...	Manual Tasks

Oracle Application Server Web Cache	▪ Locates `webcache.xml` and `internal.xml` in the source Oracle home.
	▪ Moves configuration data from *<source_MT_OH>*`/webcache/webcache.xml` to *<destination_MT_OH>*`/webcache/webcache.xml`.
	▪ Copies error pages and wallet files from the source Oracle home to the destination Oracle home.
	▪ The OracleAS Upgrade Assistant upgrades wallets by copying them from the source Oracle home to the destination Oracle home. Wallets outside of the source Oracle home need not be copied.
	
	Oracle Application Server Web Cache can have multiple listening ports, and each port can have a different wallet. When connecting to the origin server, it can use another wallet (`OSWALLET` in the example below).
	
	`<LISTEN IPADDR="ANY" PORT="4445" PORTTYPE="NORM" SSLENABLED="SSLV3_V2H">`
	`<WALLET><destination_MT_OH>/webcache/wallets/subdir1</WALLET>`
	`</LISTEN>`
	`<LISTEN IPADDR="ANY" PORT="4447" PORTTYPE="NORM" SSLENABLED="SSLV3_V2H">`
	`<WALLET>/some/other/path/wallets/default</WALLET>`
	`</LISTEN>`
	`.....`
	`.....`
	`<OSWALLET><destination_MT_OH>/webcache/wallets/default</OSWALLET>`
	
	In this example, Oracle Application Server Web Cache is using three wallets. The first and third are in the source Oracle home. The first wallet will be copied to *<destination_MT_OH>*`/webcache/wallets/subdir1`. The third wallet will be copied to *<destination_MT_OH>*`/webcache/wallets/default`. The second wallet will not be copied, since it does not reside in the Oracle home. After upgrade, the wallet in `webcache.xml` will be pointing to the original directory.
	▪ **Note:** If you customized the directory location of the event log (specified by `ACCESSLOG LOGDIR` property in the `webcache.xml` file) in Oracle9*i*AS Release 2 (9.0.2), be aware that this customization will not be upgraded.
	▪ **Note:** Web sites are often configured with Web Cache as the first listener; in these cases, the Oracle HTTP Server's Listen port may need to be synchronized with equivalent Web Cache port values after upgrade. The settings are shown in Table 3–1, "Oracle HTTP Server and Oracle Application Server Web Cache Port Settings" on page 3-9.
		Section 3.3, "Completing the OracleAS Web Cache Upgrade" on page 3-10

mod_plsql	▪ Locates `dads.conf` and `cache.conf` in the source Oracle home. ▪ Parses each source item, keeping results in memory. ▪ Applies the parsing results to the destination Oracle home. ▪ If necessary, uses the default value (*<destination_MT_OH>*`/Apache/modplsql/cache/`) for the `PlsqlCacheDirectory` property in the `cache.conf` file. ▪ Copies the `oradav.conf` file from the source Oracle home to the destination Oracle home.	None required.

Table 1–2 Components Upgraded by the OracleAS Upgrade Assistant (in processing order) and Possible Manual Upgrade Tasks

Component	The OracleAS Upgrade Assistant...	Manual Tasks
Oracle Enterprise Manager	▪ Examines the <*source_MT_OH*>sysman/emd/targets.xml file for port entries related to the targets in the file. ▪ Replaces the corresponding port entries in the <*destination_MT_OH*>sysman/emd/targets.xml file.	Section 3.5.2, "Integrating OracleAS Reports Services with Oracle Enterprise Manager"
Oracle Application Server Forms Services	▪ Locates the following files: <*source_MT_OH*>/forms90/ftrace.cfg <*source_MT_OH*>/forms90/java/oracle/forms/registry/Registry.dat User-defined Oracle Application Server Forms Services *.htm files used by formsweb.cfg <*source_MT_OH*>/forms90/search_replace.properties <*source_MT_OH*>/forms90/converter.properties <*source_MT_OH*>/forms90/server/formsweb.cfg <*source_MT_OH*>/forms90/server/default.env <*source_MT_OH*>/forms90/server/forms90.conf <*source_MT_OH*>/j2ee/properties/oc4j_bi_forms.properties (Oracle Application Server Forms Services deployment entries) User-defined Oracle Application Server Forms Services configuration files that are equivalents of formsweb.cfg and default.env ▪ Makes a backup file with a preUpgrade suffix of the files below and Copies the files from the source Oracle home to the destination Oracle home without modification. <*source_MT_OH*>/forms90/ftrace.cfg <*source_MT_OH*>/forms90/java/oracle/forms/registry/Registry.dat User-defined Oracle Application Server Forms Services *.htm files used by formsweb.cfg ▪ Copies the files below from the source Oracle home to the destination Oracle home, appending a 902 suffix to each file: <*destination_MT_OH*>/forms90/search_replace.properties <*destination_MT_OH*>/forms90/converter.properties ▪ Makes a backup of the files below with a preUpgrade suffix, and extracts customizations from the file in the source Oracle home and merges it into the file in the destination Oracle home. <*source_MT_OH*>/forms90/server/formsweb.cfg <*source_MT_OH*>/forms90/server/default.env <*source_MT_OH*>/forms90/server/forms90.conf <*source_MT_OH*>/j2ee/properties/oc4j_bi_forms.properties (Oracle Application Server Forms Services deployment entries) User-defined Oracle Application Server Forms Services configuration files that are equivalents of formsweb.cfg and default.env	Section 3.4, "Completing the Oracle Application Server Forms Services Upgrade" on page 3-12
Oracle Application Server Reports Services	▪ Copies all configuration files except jdbcpds.conf from the source Oracle home to the destination Oracle home. ▪ Copies all resource files from the source Oracle home to the destination Oracle home. ▪ Copies <*source_MT_OH*>/bin/reports.sh to the destination Oracle home, and adds the NLS_LANG environment variable, if it is not in the script. ▪ Copies reports server persistent files from the source Oracle home to the destination Oracle home.	Section 3.5, "Completing the Oracle Application Server Reports Services Upgrade" on page 3-13

1.2 Stopping OracleAS Instances

Stop all processes in the source and destination Oracle homes with these commands shown below.

- In the source Oracle home:
  1. Stop the Oracle Enterprise Manager Application Server Control:

     <*source_MT_OH*>/bin/emctl stop

  2. Stop OPMN and processes managed by it with this command:

     <*source_MT_OH*>/opmn/bin/*opmnctl stopall*

  3. Stop OracleAS Web Cache with this command:

     <*source_MT_OH*>/bin/*webcachectl stop*

  4. Stop all other running processes in the source Oracle home.
- In the destination Oracle home:
  1. Stop the Oracle Enterprise Manager Application Server Control with this command:

     <*destination_MT_OH*>/bin/emctl stop iasconsole

  2. Stop OPMN and processes managed by it with this command:

     <*destination_MT_OH*>/opmn/bin/*opmnctl stopall*

  3. Stop all other running processes in the destination middle tier Oracle home.

1.3 Using the OracleAS Upgrade Assistant

This section provides instructions for using the graphical user interface or the command-line version to perform an upgrade, and describes properties you can configure to specify logging behaviors for the OracleAS Upgrade Assistant. These topics are included:

Section 1.3.1, "Specifying Logging Behaviors for the OracleAS Upgrade Assistant"

Section 1.3.2, "Starting the OracleAS Upgrade Assistant To Use Multiple Oracle Universal Installer Inventory Locations"

Section 1.3.3, "Performing an Upgrade with the OracleAS Upgrade Assistant (Graphical User Interface (GUI) Version)"

Section 1.3.4, "Performing an Upgrade with the OracleAS Upgrade Assistant (Command-line Version)"

1.3.1 Specifying Logging Behaviors for the OracleAS Upgrade Assistant

You can configure the logging behavior of the OracleAS Upgrade Assistant by setting properties in the `<destination_MT_OH>/upgrade/iasua.properties` file. The logging properties and their uses are:

- **log.level** — Use this property to specify the level of logging for the OracleAS Upgrade Assistant and all component plug-ins. For example, `log.level=NOTIFICATION` would set the logging level for all components upgraded by the OracleAS Upgrade Assistant to NOTIFICATION.
- **<plug-in name>.log.level** — Use this property to specify the level of logging for a specific component plug-in, used to override the log.level property for a given

component upgrade. For example, `OC4J.log.level=TRACE` would set the logging level for the Oracle Application Server Containers for J2EE upgrade to TRACE, even if the log level for the OracleAS Upgrade Assistant was set to NOTIFICATION.

- **log.append** — Use this property to specify whether to append log entries to the existing log file or create a new log file. For example, `log.append=TRUE` would append log entries to the existing log file. (TRUE is the default.)

> **Note:** Property names are case sensitive. Property values are case insensitive.

Table 1–3 Logging Properties for the OracleAS Upgrade Assistant

Property Name	Description	Valid Values
log.level	Level of logging for the OracleAS Upgrade Assistant and all component plug-ins	WARNING NOTIFICATION ERROR TRACE DEBUG OFF INTERNAL_ ERROR
<plug-in name>.log.level	Level of logging for a specific component plug-in, used to override the log.level property for a given component upgrade.	OPMN OHS Web Cache OC4J modplsql Oracle Enterprise Manager Forms Reports
log.append	Specifies whether to append log entries to the existing log file or create a new log file.	TRUE FALSE

1.3.2 Starting the OracleAS Upgrade Assistant To Use Multiple Oracle Universal Installer Inventory Locations

The Oracle Universal Installer creates an inventory file, `/var/opt/oracle/oraInst.loc`, (or `etc/orainst.loc` on Linux and AIX) when it installs Oracle products. This file contains the location (full path) of the Universal Installer inventory directory, and the group name of the user who installed it. The OracleAS Upgrade Assistant populates its source Oracle home drop-down list with the information from this directory, the default inventory. Additional inventories are sometimes created after installation for the purpose of managing Oracle homes independently (thereby circumventing the Oracle Universal Installer features that track all Oracle homes in a single inventory).

If there are multiple inventory location files on the computer on which you are performing an upgrade, and you want to be able to select a non-default inventory

location, you must start the OracleAS Upgrade Assistant with the `-invptrloc` argument, specifying the inventory location file(s) for the Oracle homes involved in the upgrade. The syntax for starting the OracleAS Upgrade Assistant for multiple inventory locations is provided below.

Graphical User Interface (GUI) Version:

```
iasua.sh [[-invptrloc <Oracle Universal Installer inventory
pointer file>]...]
```

Command-line Version:

```
iasua.sh -sourcehome <9.0.2 Oracle home path> [[-invptrloc
<Oracle Universal Installer inventory pointer file>]...]
[-verbose] [-noprompt]
```

1.3.3 Performing an Upgrade with the OracleAS Upgrade Assistant (Graphical User Interface (GUI) Version)

This section provides step-by step instructions for using the OracleAS Upgrade Assistant GUI version to perform an upgrade.

1. Start the OracleAS Upgrade Assistant with the command:

   ***<destination_MT_OH>*/upgrade/iasua.sh**

   > **Note:** Use the `-invptrloc` argument, described in Section 1.3.2, "Starting the OracleAS Upgrade Assistant To Use Multiple Oracle Universal Installer Inventory Locations" if there are multiple inventories involved in the upgrade.

2. The Welcome screen appears. Click **Next**.
3. The Oracle Homes screen appears. The Source Oracle Home drop-down list contains the names of Release 2 (9.0.2) and Release 2 (9.0.3) Oracle homes found in the inventory of Oracle products on the current computer. The destination Oracle home is the 10*g* (9.0.4) Oracle home in which the OracleAS Upgrade Assistant is running.

   Select the source Oracle home from the drop-down list. Then click **Next**.
4. The Pre-Upgrade Requirements screen appears. Ensure that all requirements are fulfilled, and check the box for each. The **Next** button is active only when all boxes are checked.
5. Click **Next**. The Examining Components dialog box appears. The OracleAS Upgrade Assistant examines each component in the source Oracle home to determine whether it needs to be upgraded. The Status column for each component contains one of the following:

Table 1–4 OracleAS Upgrade Assistant Component Examination Status

Status	Meaning
in progress...	The OracleAS Upgrade Assistant is examining the component's upgrade items.
pending...	The component will be examined when the OracleAS Upgrade Assistant finishes examining the current component.
succeeded	All of the component's upgrade items are valid for upgrade.

Table 1–4 OracleAS Upgrade Assistant Component Examination Status

Status	Meaning
failed	The component has upgrade items that are missing or did not meet upgrade criteria. The OracleAS Upgrade Assistant cannot upgrade the component.

6. If one or more components failed, the Examination Failure Warning dialog box appears. Continue with Step 7.

   If all components succeeded, the Summary screen appears. Continue with Step 8.

7. Do one of the following:

   - Remedy all conditions that caused the examination to fail, using the instructions in Section 2.1, "Resolving Errors". Then, select the **Retry** option and click **OK**.
   - Select the **Continue with an incomplete upgrade** option and click **OK**.
   - Select the **Specify a different source Oracle home** option, click **OK**, then return to Step 3.
   - Select the **Cancel the upgrade process** option and click **OK**. The OracleAS Upgrade Assistant stops.

     If the examination was successful, or you chose to continue with an incomplete upgrade, the Summary screen appears.

8. The OracleAS Upgrade Assistant Summary screen appears. You can scroll to view the components, clicking the plus symbol (+) to expand a component's upgrade items. Review the components, then click **Finish**.

   Note: The Summary screen is the last screen before upgrade processing begins. Before you click Finish, verify that the choices on previous screens are correct and the upgrade items listed are ready to upgrade.

9. The Upgrading screen appears. The Status column for each component contains one of the following:

Table 1–5 OracleAS Upgrade Assistant Upgrading Status

Status	Meaning
in progress...	The OracleAS Upgrade Assistant is upgrading the component's upgrade items.
pending...	The component will be upgraded when the OracleAS Upgrade Assistant finishes upgrading the current component.
succeeded	The component was upgraded successfully.
failed	The OracleAS Upgrade Assistant could not upgrade the component.

10. After the upgrade completes, the Upgrade Failed or Upgrade Succeeded screen appears. Do one of the following.

    - Click **OK** to close the Upgrade Failure screen and remedy the conditions that prevented the components from being upgraded. Start the OracleAS Upgrade Assistant again.

See Also: Section 2.1, "Resolving Errors" and Section 2.2, "Restarting the OracleAS Upgrade Assistant".

- Click **OK** to close the Upgrade Succeeded screen.

The Upgrade Succeeded screen specifies the location of the upgrade log file and lists the post-upgrade tasks to be performed for various components.

1.3.4 Performing an Upgrade with the OracleAS Upgrade Assistant (Command-line Version)

This section explains how to start and use the OracleAS Upgrade Assistant command-line version to perform an upgrade.

Note: The OracleAS Upgrade Assistant examines components differently in the command-line version and the GUI version.

If the examination of a component fails in the command line version, then the upgrade is not performed.

If the examination of a component fails in the GUI version, the following choices are provided: retry, continue with an incomplete upgrade, specify another Oracle home, or cancel the upgrade.

---

1. Start the OracleAS Upgrade Assistant with the command:

   ***<destination_MT_OH>*/upgrade/iasua.sh -sourcehome <*source_MT_OH*>**

Note: The argument `-sourcehome` is required to start the command-line version of the OracleAS Upgrade Assistant (`iasua.sh` without this argument starts the GUI version). You can also use the following optional arguments when starting the command-line version:

`-verbose` to output detailed information to the screen during upgrade

`-noprompt` to turn off prompting and user confirmation during upgrade (by default, prompting and user confirmation are on)

Note: Use the `-invptrloc` argument, described in Section 1.3.2, "Starting the OracleAS Upgrade Assistant To Use Multiple Oracle Universal Installer Inventory Locations" if there are multiple inventories involved in the upgrade.

A prompt appears listing all the pre-upgrade requirements and asking you to verify that they have been met.

```
Validating Oracle homes

Validating component plug-ins

Initializing component plug-ins

Pre-upgrade requirements:
```

```
The destination Oracle home has not been modified since it
was installed

The source and destination OracleAS instances are not
running

The source and destination Oracle Enterprise Manager
processes are not running

Verify that each of the pre-upgrade requirements above
have been met.

Have the pre-upgrade requirements been met?[No]Yes
```

2. Ensure that all the listed requirements are met. Then answer the prompt `[Y]` Yes to continue.

   Messages similar to the following appear (The messages vary according to components found in the Oracle home):

```
Examining component "Oracle Process Manager and
Notification Server (OPMN)"

Examining component "Oracle Application Server Containers
for J2EE (OC4J)"

Examining component "Oracle HTTP Server"

Examining component "OracleAS Web Cache"

Examining component "Oracle mod_plsql"

Examining component "Oracle Enterprise Manager"

Examining component "OracleAS Forms Services"

Examining component "OracleAS Reports Services"

Upgrading component "Oracle Process Manager and
Notification Server (OPMN)"

Upgrading component "Oracle Application Server Containers
for J2EE (OC4J)"

Upgrading component "Oracle HTTP Server"

Upgrading component "OracleAS Web Cache"

Upgrading component "Oracle mod_plsql"

Upgrading component "Oracle Enterprise Manager"

The command completed successfully
```

3. If any error messages are displayed in Step 2, correct the errors as explained in Section 2.1, "Resolving Errors". Then restart the Upgrade Assistant and perform the upgrade process again.

2 Troubleshooting

This chapter contains information you need to know about troubleshooting and resolving errors that may occur during an upgrade.

2.1 Resolving Errors

If errors occur at either stage of the upgrade process, you must correct the conditions that caused them before you try the upgrade again. This section contains the following topics:

Section 2.1.1, "Resolving Common Errors"

Section 2.1.2, "Examining the Log File"

Section 2.1.3, "Reasons for Oracle Application Server Containers for J2EE Upgrade and Deployment Failures"

2.1.1 Resolving Common Errors

Under certain conditions, the OracleAS Upgrade Assistant cannot perform an upgrade. Among these are that the starting configuration is unsupported, or processes are running in the Oracle homes.

This section identifies each condition and its cause(s), and explains how to resolve it.

2.1.1.1 Source Oracle Home Not Provided by OracleAS Upgrade Assistant

If the source Oracle home does not appear as expected in the drop-down list of the Oracle Homes screen when you execute the OracleAS Upgrade Assistant, suspect one of these conditions: wrong installation type, Oracle homes are on different computers, or the Oracle home is not identified in the default inventory. The solution for each of these is detailed below.

Wrong Installation Type

The source Oracle home will not appear if the source middle tier instance is not of the same installation type as the destination middle tier instance. If this is the case, you must reinstall the destination middle tier with the same installation type as the source middle tier.

Oracle Homes on Different Computers

Another case in which the source middle tier will not appear as a selection is that the source middle tier instance is installed on a different computer from the destination middle tier instance. If this is the case, you must install the destination middle tier instance on the same computer as the source instance to be upgraded.

Oracle Home Not in Default Inventory

The OracleAS Upgrade Assistant uses the default inventory location to populate the drop-down list in the Oracle Homes screen. If the source Oracle home is not listed in the default Oracle Universal Installer inventory, then you need to provide the inventory file location to the OracleAS Upgrade Assistant. Start the OracleAS Upgrade Assistant with the `-invptrloc` option, described in Section 1.3.2, "Starting the OracleAS Upgrade Assistant To Use Multiple Oracle Universal Installer Inventory Locations" to specify the inventory location.

2.1.1.2 Upgrade Fails During OPMN, OC4J, or Oracle HTTP Server Upgrade

If the upgrade fails during the OPMN, OC4J or Oracle HTTP Server upgrade, it is probably because OPMN is still running in one or both instances (source and destination). You must stop OPMN before starting the OracleAS Upgrade Assistant. Follow the instructions in Section 1.2, "Stopping OracleAS Instances".

2.1.2 Examining the Log File

You can examine the `<destination_MT_OH>/upgrade/log/iasua.log` file and Table 2–1, "OracleAS Upgrade Assistant Error Messages" to determine the cause of examination and upgrade failures.

Note: By default, the OracleAS Upgrade Assistant logging function appends, so you should always look for the last instance of a message in the file. You can set `log.append=FALSE` in `<destination_MT_OH>/upgrade/iasua.properties` to overwrite entries instead of appending them.

2.1.2.1 Recovering From Examination Failures

To determine the cause of an examination failure:

1. Note the name of the failed component in the OracleAS Upgrade Assistant dialog or command-line output.
2. Open `<destination_MT_OH>/upgrade/log/iasua.log`.
3. Search for the message `Starting to examine component_name`.
4. Investigate the messages between the `Starting...` message and the message `Finished examining component_name with status: Failure.`

2.1.2.2 Recovering From Upgrade Failures

To determine the cause of an upgrade failure:

1. Note the name of the failed component in the OracleAS Upgrade Assistant dialog or command-line output.
2. Open `<destination_MT_OH>/upgrade/log/iasua.log`.
3. Search for the message `Starting to upgrade component_name`.
4. Investigate the messages between the `Starting...` message and the message `Finished upgrading component_name with status: Failure`.

Table 2–1 OracleAS Upgrade Assistant Error Messages

Component	Message	Possible Cause and Solution
All	`Unable to upgrade file` *`filename`*.	The file was not found in the source Oracle home, or you do not have sufficient permissions to copy the file. Determine the permissions for the file in the source Oracle home and the destination Oracle home, and adjust them as necessary.
Oracle Application Server Containers for J2EE	`J2eeDeploymentExceptio n`	An application EAR file is not 100% J2EE compliant. Use the `validateEarFile` utility to identify the noncompliant characteristics, and correct them. Instructions for using the utility are provided in Section 2.1.3.2, "Application Deployment and J2EE Compliance Requirements".
Oracle Application Server Forms Services	`Save files operation failed.`	The copy operation failed. Some files are copied "as is" from <*source_MT_OH*> (i.e., registry.dat and ftrace.cfg). Verify that all of these files exist and that permissions and disk space are sufficient for a copy operation.
Oracle Application Server Forms Services	`Invalid section in the <formsweb.cfg> <default.env> file.`	There is an invalid entry in the named file in <*source_MT_OH*>. Examine the file, and locate and correct any errors.
Oracle Application Server Forms Services	`Invalid or missing configuration file.`	There is an invalid configuration file in <*source_MT_OH*>. Examine the file, and locate and correct any errors.
Oracle Application Server Forms Services	`Invalid or missing Forms configuration file <file name>.`	The Upgrade Assistant is unable to locate the configuration files specified in the `formsweb.cfg` file (`*htm` and `*env` files), or the user-defined FormsServlet configuration file specified in `oc4j_bi_forms.properties`. Ensure that all files specified in the entries are valid and exist in the specified location.

Table 2–1 OracleAS Upgrade Assistant Error Messages

Component	Message	Possible Cause and Solution
Oracle Application Server Forms Services	`Forms is not configured in the Source Oracle Home <version number>, Forms upgrade cannot proceed.`	If Forms services are not configured in the source OracleAS middle tier installation, then the Upgrade Assistant will not upgrade Oracle Application Server Forms Services. Ignore this message; if Oracle Application Server Forms Services is not configured in the source Oracle home, then upgrade is unnecessary.
Oracle Application Server Forms Services	`Forms is not configured in the Destination Oracle Home <version number>, Forms upgrade cannot proceed.`	If Forms services are not configured in the destination middle tier installation, then the Upgrade Assistant will not upgrade Oracle Application Server Forms Services. Configure Oracle Application Server Forms Services in the destination Oracle home.

Oracle HTTP Server (mod_plsql)	`java.io.FileNotFoundException...Apache/modplsql/conf/dads.conf`
	or
	`java.io.FileNotFoundException...Apache/modplsql/conf/cache.conf`
		The file was not found. Provide a file at the location specified.

2.1.3 Reasons for Oracle Application Server Containers for J2EE Upgrade and Deployment Failures

This section discusses reasons for which an Oracle Application Server Containers for J2EE upgrade may fail.

2.1.3.1 Configuration Change Requirements

If a configuration does not perform as expected after an upgrade, it might be because configuration changes were made to OC4J application files by means other than the Oracle Enterprise Manager Application Server Control. Only the changes made by the Oracle Enterprise Manager Application Server Control will be included in the OC4J upgrade performed by the OracleAS Upgrade Assistant. Manually edited files may not be in the scope of the managed configuration, and the edits may not be preserved in an upgrade.

If you use Distributed Configuration Management's dcmctl utility to perform configuration changes, see the *Distributed Configuration Management Reference Guide* for instructions and a complete discussion on the correct usage of the commands.

2.1.3.2 Application Deployment and J2EE Compliance Requirements

OC4J deployment enforces J2EE compliance rules, so the OracleAS Upgrade Assistant may not upgrade applications that are not fully J2EE compliant. The OracleAS Upgrade Assistant simply reads the files in the source Oracle home and attempts to deploy them to the destination Oracle home; if deployment fails, it could be because an application is not J2EE compliant. If the OracleAS Upgrade Assistant cannot deploy an application for any reason, it logs the exception in the <destination_MT_OH>/upgrade/log/iasua.log. The exception may not be explicitly described as a J2EE compliance issue, but that may be the reason for the failure. Knowledge of the J2EE and EJB specifications, and the EJB features used in applications will be helpful in

preventing and troubleshooting deployment failures (10*g* (9.0.4) supports a higher version of the EJB specification than Release 2 (9.0.2)).

While the development of J2EE applications is standardized and portable, the XML configuration files are not. Multiple XML files may need to be configured for an OC4J application to be deployed, and the required configuration varies according to the services the application uses. For example, if the application uses a database, the DataSource object in the data-sources.xml file must be configured.

Validating EAR Files for J2EE Compliance

The `dcmctl` utility provides a J2EE compliance validation command. It takes one input, the name of an EAR file, and then lists non-compliant characteristics of that file. The syntax is:

```
<destination_MT_OH>/dcm/bin/dcmctl validateEarFile -f  <full
path and filename for ear file>
```

You must provide the full path to the EAR file.

If you connect to the Internet using a proxy server, you must configure proxy settings so that the validation routine can access DTDs (for example, on the Sun Microsystems site). To do this, you define an environment variable called `ORACLE_DCM_JVM_ARGS`, which specifies a hostname and port for the proxy. For example, using `tcsh`, the command is:

```
setenv ORACLE_DCM_JVM_ARGS
"-DhttpProxy.host=www-proxy.hostname.com -DhttpProxy.port=9999"
```

where hostname is the host name and 9999 is the port number. The method of defining this environment variable depends on the platform, so refer to system documentation for instructions on defining this variable.

If there is no firewall to connect to an external network, use the `-noproxy` flag with the command. For example:

```
<destination_MT_OH>/dcm/bin/dcmctl validateEarFile -f <full path
and filename for ear file> -noproxy
```

Example 2–1 validateEarFile Command and Output for J2EE-Compliant Application

```
dcmctl validateEarFile -v -f simple.ear
No J2EE XML/DTD validation errors were found
```

Example 2–2 validateEarFile Command and Output for non-J2EE-Compliant Application

```
dcmctl validateEarFile -v -f petstore.ear
Warning: J2EE/DTD validation errors were found
ADMN-906001   {0}   Base Exception:
oracle.ias.sysmgmt.deployment.j2ee.exception.J2eeDeploymentException:Cannot get
xml   document by parsing /var/tmp/jar50152.tmp: Invalid element 'servlet' in
content of 'web-app',   expected elements '[servlet-mapping, session-config,
mime-mapping, welcome-file-list,   error-page, taglib, resource-ref,
security-constraint, login-config, security-role, env-entry,   ejb-ref]'.
```

It is a good idea to review all applications for overall J2EE compliance before upgrading them, since there are cases in which an application is deployable, but delivers unpredictable or undesirable server behavior. For example, ensure that each application has a unique context root defined in `application.xml`.

2.2 Restarting the OracleAS Upgrade Assistant

You can restart the OracleAS Upgrade Assistant after it has partially or completely processed an Oracle home. Follow these steps:

1. Start the OracleAS Upgrade Assistant GUI version as described in "Performing an Upgrade with the OracleAS Upgrade Assistant (Graphical User Interface (GUI) Version)", or the command-line version as described in "Performing an Upgrade with the OracleAS Upgrade Assistant (Command-line Version)".

   The OracleAS Upgrade Assistant displays one of the following messages, depending on the outcome of the previous upgrade:

   If the previous upgrade was unsuccessful, then the message is:

   ```
   The OracleAS Upgrade Assistant has already processed this
   destination Oracle home directory, it didn't complete
   successfully.
   ```

   If the previous upgrade was successful, then the message is:

   ```
   The OracleAS Upgrade Assistant has already successfully
   processed this destination Oracle home directory.
   ```

2. Close the dialog and continue with the upgrade.

3

Completing the Upgrade

This chapter describes tasks you may need to perform to complete the upgrade after the OracleAS Upgrade Assistant has finished processing. Some or all of these may be necessary, depending on the configuration upgraded. This section discusses the following topics:

3.1 Completing the Oracle Application Server Containers for J2EE (OC4J) Upgrade

The OracleAS Upgrade Assistant performs many of the Oracle Application Server Containers for J2EE (OC4J) upgrade tasks. However, some components of OC4J may require manual adjustments, or may have characteristics of which you should be aware before using OracleAS 10*g* (9.0.4).

This section details upgrade considerations for some sub-components of OC4J. If you use Oracle JMS, Oracle JDBC, the XML Parser for JAXP/XDK, or Oracle JSP pages, some or all of the topics in this section may be useful to you.

3.1.1 Upgrading Oracle Application Server Java Authentication and Authorization Service (JAZN) LDAP Security Settings

The OracleAS Upgrade Assistant does not upgrade the JAZN settings (the `orion-application.xml` file). Therefore, if you upgraded OC4J applications that use the JAZN LDAP User Manager for security, to complete the upgrade, you must perform the following steps:

1. Using the Oracle Enterprise Manager Application Server Control, in the **General Properties** section of the OC4J application, under **User Manager**, select the **JAZN LDAP User Manager**.
2. In the **Security Settings** section of the OC4J application, under **Security Roles**, map **Users/Groups** to the same role defined in the source Oracle home.

3.1.2 Upgrading JAZN Library Path Entries

In Oracle Application Server 10*g* (9.0.4), the `jazn.jar` file has been split into two JAR files: `jazn.jar` and `jazncore.jar`. For this reason, after upgrading OC4J applications that use JAZN, both JAR file names must have library path entries in the `application.xml` file.

Ensure that the `application.xml` file contains both of the entries below:

```
<library path="904 J2EE HOME/jazn.jar"/>

<library path="904 J2EE HOME/jazncore.jar"/>
```

where

```
<904 J2EE HOME> = <destination_MT_OH>/j2ee/home
```

3.1.3 Upgrading OC4J Instances Created by the Installer

Customizations that were made to OC4J instances in the file must be upgraded manually. This includes the instances created by the installer (home, OC4J_WIRELESS, OC4J_DEMOS, OC4J_PORTAL OC4J_BI_FORMS). The OracleAS Upgrade Assistant upgrades customizations to OC4J instances that were created by the user.

3.1.4 Upgrading application.xml Entries

If you have customized entries in the application.xml file, such as library paths, Java options, and OC4J options, you must upgrade them manually.

3.1.5 Upgrading the jms.xml File

The jms.xml file is not automatically upgraded from earlier versions. All queues, topics, and connection factories defined in the jms.xml file in the source Oracle home must be added to the jms.xml file in the destination Oracle home.

3.1.6 Using the Compatibility Test Suite (CTS) Compatibility Flag for Backward Compatibility

In Oracle Application Server 10*g* (9.0.4), OC4J by default complies with the J2EE 1.3 specification. In some cases, this results in behavior that differs from that seen with previous OC4J implementations. To allow for backward compatibility, OC4J supports a CTS compliance flag that you can set to false to revert to previous OC4J behavior in the following components:

- Oracle JMS
- Oracle JDBC
- Oracle XML parser for JAXP/XDK

The compliance behavior of OC4J is determined by the flag `oracle.cts.useCtsFlags`, with a default value of `true`. If any of the upgrade issues are critical in a particular application, you can disable CTS compliance and revert to old behavior for an OC4J instance by setting the flag value to false in an OC4J properties file, and providing the location of the properties file to OC4J.

For example, the file *<destination_MT_OH>*`/j2ee/home/config/oc4j.properties` might contain the flag:

```
oracle.cts.useCtsFlags=false
```

Supply the name and location of a properties file to OC4J through an `<oc4j-option>` element in the *<destination_MT_OH>*`/opmn/conf/opmn.xml` file, as in the following example:

Example 3–1 oc4j-option Element in opmn.xml File

```
<oc4j>
...
 <oc4j-option value="-p <destination_MT_OH>/j2ee/home/config/oc4j.properties"/>
...
</oc4j>
```

This is equivalent to starting OC4J as follows in standalone mode (where % is a system prompt):

Example 3–2 Starting OC4J in Standalone Mode

```
% java -jar oc4j.jar -p <destination_MT_OH>/j2ee/home/config/oc4j.properties
```

3.1.6.1 CTS Compatibility and OJMS

In the OracleAS 10*g* (9.0.4) implementation of Oracle JMS (OJMS), which complies with J2EE 1.3, some behavior differs from OJMS behavior in Oracle9iAS Release 1 (1.0.2.2). (There are no such upgrade considerations between Oracle9iAS releases 9.0.2 and 9.0.3.) The differences are as follows:

- JMSExpiration—In the OJMS 10*g* (9.0.4) J2EE 1.3-compliant implementation, the JMSExpiration header value in a dequeued message is the sum of the JMS timestamp when the message was enqueued, and the time-to-live. This value is expressed in milliseconds from midnight, January 1, 1970 to the current Greenwich Mean Time. If a message never expires, the value is 0.

  In the OJMS 1.0.2.2 implementation, the JMSExpiration header value in a dequeued message is the duration until expiration of the message, in milliseconds. If a message never expires, the value is -1.

- JMSPriority—In the OJMS Release 2 (9.0.4) 1.3-compliant implementation, 9 is the highest priority, 0 is the lowest priority, and 4 is the default priority.

  In the OJMS 1.0.2.2 implementation, `java.lang.Integer.MIN_VALUE` is the highest priority, `Integer.MAX_VALUE` is the lowest priority, and 1 is the default priority.

- Durable subscribers—In the OJMS 10*g* (9.0.4) J2EE 1.3-compliant implementation, durable Topic Subscribers with the same name are not allowed under any circumstances.

  In the OJMS 1.0.2.2 implementation, durable Topic Subscribers with the same name are allowed if they are subscribed to different topics.

- Strongly typed JMS selectors—In accordance with the JMS 1.02b specification and J2EE 1.3 compliance requirements, the OJMS 10*g* (9.0.4) implementation uses only a certain subset of SQL92 syntax for selector expression syntax, with the following mandated restrictions:

  - Selector expressions are strongly typed, meaning operators and operands in arithmetic comparisons must be of the same type. Automatic type conversions for the purpose of comparison, such as converting the string "1" to the integer 1, are prohibited.
  - String and boolean comparisons are restricted to "=", "<", and ">". Two strings are equal only if they contain the exact same sequence of characters.
  - The "!=" operator is prohibited.

  The OJMS 1.0.2.2 implementation is not subject to these restrictions or to the limited subset of SQL92 syntax for selector expression syntax.

3.1.6.2 CTS Compatibility and JDBC

In the OracleAS 10*g* (9.0.4) implementation of Oracle JDBC, which complies with J2EE 1.3, some behavior differs from JDBC behavior in Oracle9iAS Release 2 (9.0.2) and prior. The differences are as follows:

- Java types for NUMBER columns—In 10*g* (9.0.4), the `getObject()` method of a result set (`java.sql.ResultSet` instance) returns a `java.lang.Double` value for a NUMBER column with precision, or a `java.math.BigDecimal` value for a NUMBER column without precision.

  In Release 2 (9.0.2) and prior releases, `getObject()` returns a BigDecimal value for any NUMBER column.

- Metadata for `NUMBER` columns—In 10*g* (9.0.4), the `getColumnTypeName()` method of a result set metadata object (`java.sql.ResultSetMetaData` instance) returns `"FLOAT"` for a `NUMBER` column with precision, or `"NUMBER"` for a `NUMBER` column without precision. The `getColumnType()` method returns `java.sql.Types.FLOAT` for a `NUMBER` column with precision, or `Types.NUMBER` for a `NUMBER` column without precision.

  In Release 2 (9.0.2) and prior releases, `getColumnTypeName()` returns `"NUMBER"` for any `NUMBER` column, and `getColumnType()` returns `Types.NUMBER` for any `NUMBER` column.

- Java types for DATE and TIMESTAMP columns—In 10*g* (9.0.4), the `getObject()` method of a result set returns a `java.sql.Date` value for a `DATE` column, and a `java.sql.Timestamp` value for a `TIMESTAMP` column.

  In Release 2 (9.0.2) and prior releases, `getObject()` returns a `java.sql.Timestamp` value for a `DATE` column. (`TIMESTAMP` columns were not supported.)

- Exceptions for inappropriate SQL statements—In 10*g* (9.0.4), if an `executeQuery()` call in a statement object contains anything but a `SELECT` statement (such as if it instead contains an `INSERT` or `UPDATE` statement), the JDBC driver properly throws an exception. Similarly, if an `executeUpdate()` call

contains a `SELECT` statement, the driver properly throws an exception. (An `UPDATE`, `INSERT`, or `DELETE` statement is expected.)

In Release 2 (9.0.2) and prior releases, these situations did not result in exceptions.

3.1.6.3 CTS Compatibility and the JAXP/XDK XML Parser

In the OracleAS 10*g* (9.0.4) implementation of the XML parser for JAXP/XDK, which complies with J2EE 1.3, some behavior differs from XML parser behavior in Oracle9iAS Release 2 (9.0.2) and prior. The differences are as follows:

- `getNamespaceURI()` null return values—In 10*g* (9.0.4), the `getNamespaceURI()` method returns 'null' if the namespace is not defined for an element or attribute.

  In Release 2 (9.0.2) and prior releases, the `getNamespaceURI()` method returns `'""'` in these circumstances.

- `getLocalName()` null return values—In 10*g* (9.0.4), the `getLocalName()` method returns `'null'` if the element or attribute was created using a DOM level 1 API call to `createElement()` or `createAttribute()`.

  In Release 2 (9.0.2) and prior releases, the `getLocalName()` method returns `'"Transfer interrupted!"'` in these circumstances.

- `getPrefix()` null return values—In 10*g* (9.0.4), the `getPrefix()` method returns `'null'` if the element or attribute was created using a DOM level 1 API call to `createElement()` or `createAttribute()`.

  In Release 2 (9.0.2) and prior releases, the `getPrefix()` method returns `'""'` in these circumstances.

  Note: The `getNamespaceURI()`, `getLocalName()`, and `getPrefix()` methods exist with the above changes in the `XMLElement` and `XMLAttr` classes of the `oracle.xml.parser.v2` package.

- SAX exceptions—In 10*g* (9.0.4), registered error handlers throw a `SAXException` or `SAXParseException` in error conditions.

  In Release 2 (9.0.2) and prior releases, error handlers throw an `XMLParseException` in error conditions.

- I/O exceptions—In 10*g* (9.0.4), an `IOException` is thrown as is in I/O error conditions.

  In Release 2 (9.0.2) and prior releases, an `IOException` is wrapped in an `XMLParseException`.

3.1.7 Upgrade Considerations for Enterprise Java Beans

Beginning with Oracle9*i*AS Release 2 (9.0.3), Oracle Application Server Containers for J2EE complies with the J2EE 1.3 specification and implements the Enterprise Java Beans (EJB) 2.0 specification in entirety. Therefore, if you are upgrading from Release 2 (9.0.2) to 10*g* (9.0.4), applications using EJB features in the areas of container-managed persistence and container-managed relationships will require modification.

See Also: *Oracle Application Server Containers for J2EE Enterprise JavaBeans Developer's Guide*, Appendix C.

3.1.8 Upgrade Considerations for the OC4J Java Server Pages (JSP) Container

This section describes JSP settings that are affected by the upgrade.

3.1.8.1 Enabling Extra Imports

Beginning with Oracle9iAS Release 2 (9.0.3), the OC4J JSP container by default imports the packages listed below into any JSP page, in accordance with the JSP specification. No `page` directive import settings are required.

```
javax.servlet.*
```

```
javax.servlet.http.*
```

```
javax.servlet.jsp.*
```

In previous releases, the following packages were also imported by default:

```
java.io.*
```

```
java.util.*
```

```
java.lang.reflect.*
```

```
java.beans.*
```

For backward compatibility, you can use the JSP `extra_imports` configuration parameter as a workaround. Alternatively, you can add desired imports through `page` directives or global includes. See the *Oracle Application Server Containers for J2EE Support for JavaServer Pages Developer's Guide* for information about these topics.

3.1.8.2 Setting Additional JSP Flags for Backward Compatibility

When upgrading to Oracle Application Server 10*g* (9.0.4) and using JSP pages, use appropriate settings for the following important JSP configuration parameters.

- `check_page_scope`
- `forgive_dup_dir_attr`

These are set as initialization parameters for the JSP front-end servlet, either in the global-web-application.xml file or in the application orion-web.xml file. Here is an example:

Example 3–3 JSP Configuration Parameters for Upgrading to 10g (9.0.4)

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

See the *Oracle Application Server Containers for J2EE Support for JavaServer Pages Developer's Guide* for more information about JSP configuration parameters.

`check_page_scope` (boolean; default: `false`): This parameter was introduced in Oracle9iAS Release 2 (9.0.3). For OC4J environments, set it to `true` to enable Oracle-specific page-scope checking by the `JspScopeListener` utility.

This parameter is not relevant for non-OC4J environments. For JServ, Oracle-specific page-scope checking is always enabled. For other environments, the Oracle-specific

implementation is not used and you must use the checkPageScope custom tag for JspScopeListener page-scope functionality. See the Oracle Application Server Containers for J2EE JSP Tag Libraries and Utilities Reference for information about JspScopeListener.

`forgive_dup_dir_attr` (boolean; default: `false`): This parameter was introduced in Oracle9iAS Release 2 (9.0.3). Set it to true to avoid translation errors in a JSP 1.2 environment such as OC4J if you have duplicate settings for the same directive attribute within a single JSP translation unit (a JSP page plus anything it includes through `include` directives).

The JSP 1.2 specification directs that a JSP container must verify that directive attributes, with the exception of the page directive `import` attribute, are not set more than once each within a single JSP translation unit.

The JSP 1.1 specification did not specify such a limitation. OC4J offers the `forgive_dup_dir_attr` parameter for backward compatibility.

3.1.9 Considering JDK 1.4 Issues: Cannot Invoke Classes Not in Packages

Among the migration considerations in moving to a Sun Microsystems JDK 1.4 environment, which is the environment that is shipped with Oracle Application Server 10*g* (9.0.4), there is one of particular importance to servlet and JSP developers.

As stated by Sun Microsystems, "The compiler now rejects import statements that import a type from the unnamed namespace." (This was to address security concerns and ambiguities with previous JDK versions.) Essentially, this means that you cannot invoke a class (a method of a class) that is not within a package. Any attempt to do so will result in a fatal error at compilation time.

This especially affects JSP developers who invoke JavaBeans from their JSP pages, as such beans are often outside of any package (although the JSP 2.0 specification now requires beans to be within packages, in order to satisfy the new compiler requirements). Where JavaBeans outside of packages are invoked, JSP applications that were built and executed in an OC4J 9.0.3 / JDK 1.3.1 or prior environment will no longer work in an OC4J 9.0.4 / JDK 1.4 environment.

Until you update your application so that all JavaBeans and other invoked classes are within packages, you have the alternative of reverting back to a JDK 1.3.1 environment to avoid this issue.

Notes:

The `javac -source` compiler option is intended to allow JDK 1.3.1 code to be processed seamlessly by the JDK 1.4 compiler, but this option does not account for the "classes not in packages" issue.

Only the JDK 1.3.1 and JDK 1.4 compilers are supported and certified by OC4J. It is possible to specify an alternative compiler by adding a `<java-compiler>` element to the `server.xml` file, and this might provide a workaround for the "classes not in packages" issue, but no other compilers are certified or supported by Oracle for use with OC4J. (Furthermore, do not update the `server.xml` file directly in an Oracle9*i*AS environment. Use the Oracle Enterprise Manager Application Server Control.)

For more information about the "classes not in packages" issue and other JDK 1.4 compatibility issues, refer to the following Web site:

http://java.sun.com/j2se/1.4/compatibility.html

In particular, click the link "Incompatibilities Between Java 2 Platform, Standard Edition, v1.4.0 and v1.3".

3.1.10 Considering Modified Servlet APIs and Behavior

When upgrading to Oracle Application Server 10*g* (9.0.4) and using servlets, consider the following changes in servlet APIs and behavior:

- Changes relating to getRequestURI()
- Changes regarding filtering of servlets that are forward or include targets

3.1.10.1 Changes Relating to getRequestURI()

In previous Oracle9iAS releases, whenever Oracle HTTP Server received a request, it would unencode the URI before passing it to OC4J. Therefore, servlets making computations based on the response of `getRequestURI()` (a method on the request object) were implicitly getting a value that had been unencoded one time. As of the OC4J 9.0.4 implementation, Oracle HTTP Server will send OC4J an unaltered version of the URI, which in turn is used by OC4J as the return value of getRequestURI().

If the `mod_rewrite` module is being used in conjunction with `mod_oc4j` in communications between Oracle HTTP Server and OC4J, the rewritten URI that is sent to `mod_oc4j` is the same as what is sent to OC4J, and the return value of `getRequestURI()` will have had `mod_rewrite` rules applied to it.

The mod_rewrite and mod_oc4j modules are discussed in the Oracle HTTP Server Administrator's Guide. Further details about mod_rewrite are available in the Apache Server documentation.

3.1.10.2 Filtering of Servlets That Are Forward or Include Targets

In previous OracleAS releases, if a filtered servlet forwards to or includes another servlet, the target servlet, by default, is also filtered. In Oracle Application Server 10*g* (9.0.4), this is no longer the default behavior. Having the target servlet not filtered by default matches the intention of the servlet specification.

This behavior is configurable: in the OC4J 9.0.4 implementation, it is according to the oracle.j2ee.filter.on.dispatch environment flag (false by default); in future implementations, it will be according to web.xml configuration as set forth in the servlet 2.4 specification.

3.2 Completing the Oracle HTTP Server Upgrade

This section describes post-upgrade tasks for the Oracle HTTP Server that you may need to perform. The OracleAS Upgrade Assistant upgrades the standard settings for the Oracle HTTP Server. If you have configuration files or documents that are in non-standard locations or referenced in non-standard ways, you must upgrade these manually. These, and other configuration-specific cases for manual upgrade, are detailed below.

- **If you want the Oracle HTTP Server to listen on a port numbered lower than 1024:** The HTTP server executable `apachectl` must have root user privileges to bind to ports numbered lower than 1024. Follow these steps to grant root privileges to the executable:

  1. Log in to the root account.

2. Navigate to <*destination_MT_OH*>/Apache/Apache/bin and issue these commands:

   ```
   chown root .apachectl

   chmod 6750 .apachectl
   ```

3. Exit the root account.

- **If mod_osso was configured:** If mod_osso was configured, after upgrade, the `osso.conf` file continues to use the Release 2 (9.0.2) partner entry in the Single Sign-On server. The 10*g* (9.0.4) partner entry in the Single Sign-On server is not being used, and will cause a broken link (invalid URL) when the application logs out. You should remove the 10*g* (9.0.4) partner entry. In addition, if the name of the entry in use is obsolete (in that it refers in some way to the source Oracle home), you may wish to rename it.
- **If there are configuration files in non-default locations:** If `httpd.conf`, `mod_oc4j.conf`, `mod_osso.conf` and `moddav.conf` files are not in the default location, you must upgrade them manually by applying the customizations in the files in the source Oracle home to the files in `the destination Oracle home`.
- **If there are custom files and directories referenced by Oracle HTTP Server configuration files:** Because the OracleAS Upgrade Assistant only upgrades the items listed in Section A.1.4.1, "OHS Upgrade Items" on page A-7 of *Oracle Application Server 10g Upgrading to 10g (9.0.4)*, there could be files or directories referred to by directives such as `Alias`, `mod_rewrite`, and log directives, such as `ErrorLog`, that are not present after the upgrade. Ensure that all such items are upgraded manually and exist in the locations expected by the directives. If these files or directives are missing after the upgrade, the Oracle HTTP Server may not start. You can identify errors by starting the Oracle HTTP Server individually after the upgrade, and examining the <*destination_MT_OH*>`/Apache/Apache/logs/error_log` for errors associated with these items.
- **If there are Dynamic Monitoring Service (DMS) configuration elements in the httpd.conf and mod_oc4j.conf files**: You must relocate these configuration elements into the `dms.conf` file.
- **If Oracle Application Server Web Cache is the first listener:** If OracleAS Web Cache is configured as the first listener, ensure that the Oracle HTTP Server directives listed in Table 3–1 have the same values as the corresponding OracleAS Web Cache elements. In particular, note that the Oracle HTTP Server Port directive specifies the port number of a front-end load balancer or reverse proxy. Thus, if Oracle Application Server Web Cache is used, then the Oracle HTTP Server Port directive should have the value of the port on which OracleAS Web Cache is listening.

Table 3–1 Oracle HTTP Server and Oracle Application Server Web Cache Port Settings

Oracle HTTP Server Directive	Oracle Application Server Web Cache Element
VirtualHost	Site definitions
Listen	Origin server ports
VirtualHost, Listen	Site-to-server mappings
Port	Listen

- **If you have static documents in the default DocumentRoot directory that you want to upgrade:** The OracleAS Upgrade Assistant locates static document files

and directories for upgrade in the location specified in the `DocumentRoot` directive. The `DocumentRoot` directive defines the location for static documents and related directories. The base server has a document root location, and each virtual host has one. The OracleAS Upgrade Assistant copies files under these directories to the destination Oracle home. The default DocumentRoot directory `<source_MT_OH>/Apache/Apache/htdocs` contains demonstration programs and release notes placed there by the installer, so the OracleAS Upgrade Assistant does not upgrade this directory. You must upgrade this directory manually.

3.3 Completing the OracleAS Web Cache Upgrade

This section outlines tasks you may need to perform to complete the OracleAS Web Cache upgrade after the OracleAS Upgrade Assistant has finished processing. These tasks include enabling the use of privileged ports, and upgrading a cluster.

3.3.1 Enabling the webcached Executable to Run as The Root User

OracleAS Web Cache will not start after upgrade if the port settings of 80 and 443 were upgraded from the Oracle9*i*AS Release 2 (9.0.2) Web Cache to the Oracle Application Server 10*g* (9.0.4) Web Cache.

Because port numbers under 1024 are reserved for privileged processes in UNIX, the `webcached` executable in 10*g* (9.0.4) must run as root in order to start the cache server process and bind to these ports.

A script is provided that enables the `webcached` executable to run as the root user:

`<destination_MT_OH>/webcache/bin/webcache_setuser.sh`

Log in to the system as the root user and run the script.

See Also: *Oracle Application Server Web Cache Administrator's Guide*

3.3.2 Upgrading an OracleAS Web Cache Cluster

If you have a OracleAS Web Cache cluster, you can upgrade one cache cluster member at a time. The caches will continue to function, but because the other cluster members have a different version of the configuration, the caches will not forward requests to cache cluster members operating with a different version.

For example, if you upgrade Cache_A to the current version, but have not yet upgraded Cache_B and Cache_C, Cache_A will not forward requests to the cache cluster members Cache_B and Cache_C.

In this situation, the Operations page in Web Cache Manager indicates that the Operation Needed is Incompatible software version.

Note: When the cache cluster members are not running the same version of OracleAS Web Cache, you can still invalidate documents and you can propagate the invalidation to other cluster members, but the invalidation request must originate with the cache that is operating with 9.0.2 version of OracleAS Web Cache.

After you upgrade the cache cluster members, you must perform the following additional steps to synchronize the configuration for the members of the cluster:

1. If the caches have not been started, for each upgraded cache, start OracleAS Web Cache and OracleAS Web Cache Manager. On the command line, enter:

```
opmnctl startproc ias-component=WebCache
```

This command starts the OracleAS Web Cache cache server process and admin server process.

2. In a browser, enter the URL for the OracleAS Web Cache Manager for one of the upgraded caches, and, when prompted, enter the username and password for the ias_admin or administrator user.
3. In the navigator frame, select **Administration > Operations**.

   The **Operations** page appears.
4. In the Operations page, click **Retrieve**.

   Web Cache retrieves the cache-specific configuration information from the remote cache cluster members. Then, Web Cache Manager indicates that the Operation Needed is Propagate Configuration.
5. To propagate the configuration to all cache cluster members, select **All caches** and an **Interval** of **Immediate**. Then, click **Propagate**.
6. Restart the caches by selecting **All caches** and an Interval. Then, click **Restart**. (Note that you can perform this operation as you upgrade each cache, or you can perform this operation after all of the cache cluster members have been upgraded.)

3.3.3 Upgrading a Web Cache Cluster from Release 2 (9.0.2.x) to 10*g* (9.0.4)

A Release 2 (9.0.2.x) cache cannot accept invalidation messages from a 10*g* (9.0.4) cache. In a configuration that uses a OracleAS Web Cache cluster with a mixture of Release 2 (9.0.2.x) and 10*g* (9.0.4) cluster members, you must configure the Load Balancer to send invalidation messages only to the Release 2 (9.0.2.x) members.

When upgrading a cache cluster from Release 2 (9.0.2.x) to 10*g* (9.0.4), remove cluster members one at a time from the invalidation pool for the Load Balancer prior and upgrade them. Once all the cluster members are upgraded, add them back to the invalidation pool. As an example, assume a configuration with a Load Balancer in front of a cache cluster that is comprised of four members, webche1-host, webche2-host, webche3-host, and webche4-host, all running Release 2 (9.0.2.x).

To upgrade this cache cluster:

1. In the Load Balancer configuration, remove webche1-host from the pool that is responsible for invalidation.
2. Upgrade webche1-host from Release 2 (9.0.2.x) to 10*g* (9.0.4).
3. In the Load Balancer configuration, remove webche2-host from the pool that is responsible for invalidation.
4. Upgrade webche2-host from Release 2 (9.0.2.x) to 10*g* (9.0.4).
5. In the Load Balancer configuration, remove webche3-host from the pool that is responsible for invalidation.
6. Upgrade webche3-host from Release 2 (9.0.2.x) to 10*g* (9.0.4).
7. Upgrade webche4-host from Release 2 (9.0.2.x) to 10*g* (9.0.4). As this is the last cache member in the Load Balancer configuration, it is not necessary to remove it from the invalidation pool.
8. In the Load Balancer configuration, add webche1-host, webche2-host, and webche3-host back into the pool that is responsible for invalidation.

3.4 Completing the Oracle Application Server Forms Services Upgrade

The OracleAS Upgrade Assistant moves most of the Oracle Application Server Forms Services configuration data from the source to the destination Oracle home. However, there may be manual tasks remaining after the upgrade. This section explains how to perform these tasks.

Note: After the upgrade, the `default.env` file contains the default Oracle Application Server Forms Services environment variables and any user defined environment variables. The Upgrade Assistant upgrades any user defined environment variables to the destination Oracle Home `default.env` file. Any user modifications to the default variables in the source Oracle Home `default.env` file will be extracted and appended to the default environment variable values that the installer puts in the destination Oracle Home `default.env` file.

3.4.1 Upgrading the tnsnames.ora File

Entries may have been added to, or changed in, the `tnsnames.ora` file between the installation of Oracle9*i*AS Release 2 (9.0.2) and upgrade to 10*g* (9.0.4). If so, you must upgrade this file manually so that any added or changed entries are available in 10*g* (9.0.4).

> **See Also:** Section 3.6, "Upgrading the tnsnames.ora File" on page 3-15.

If an error occurs after upgrade, it may be because the `tnsnames.ora` file was overwritten by another component upgrade. A missing or incorrect entry yields the following error:

```
ORA-12154: TNS: Could not resolve service name.
```

3.4.2 Upgrading Forms *.fmx Files

If you have deployed these files within the source Oracle home, you must manually copy them to the same location in the destination Oracle home. If the `*.fmx` files are not under the Oracle home on the file system, then no action is needed, as the `FORMS90_PATH` will be upgraded by the OracleAS Upgrade Assistant, and it will be valid after the upgrade.

3.4.3 Upgrading User-defined Aliases for Oracle Application Server Forms Services Servlets

If you defined any aliases for the OracleAS Forms Services servlets in:

`<`*`source_MT_OH`*`>/j2ee/OC4J_BI_FORMS/applications/forms90app/forms90web/WEB-INF/web.xml`, then you must manually copy these entries to:

`<`*`destination_MT_OH`*`>/j2ee/OC4J_BI_FORMS/applications/forms90app/forms90web/WEB-INF/web.xml`.

3.4.4 Upgrading forms90app.ear Deployed in User-defined OC4J Instances

The `forms90app.ear` file is deployed by default into the OC4J_BI_Forms OC4J instance. Note that the Upgrade Assistant upgrades all user-defined OC4J instances and the applications deployed under these instances to the destination Oracle home.

Thus, if you have deployed the `forms90app.ear` file into one of the user-defined OC4J instances in the source Oracle home, the Upgrade Assistant will upgrade this deployment into the corresponding OC4J instance in the destination Oracle home.

As a result, the source Oracle home Release 2 (9.0.2) `forms90app.ear` file is deployed into the destination Oracle home. This causes the configuration of OracleAS Forms Services 10*g* (9.0.4) to be incorrect, because it requires the 10*g* (9.0.4) EAR file in order to function properly.

Therefore, you must undeploy the `forms90app.ear` file from these upgraded OC4J instances in the destination Oracle home, and then deploy `forms90app.ear` in the destination Oracle home again.

The Oracle Application Server Forms Services 10*g* (9.0.4) `forms90app.ear` file is located in: `<destination_MT_OH>/forms90/j2ee`.

3.5 Completing the Oracle Application Server Reports Services Upgrade

The OracleAS Upgrade Assistant performs most of the Oracle Application Server Reports Services upgrade. However, it does not process the script files `<source_MT_OH>/bin/rw*.sh` and the configuration file `<source_MT_OH>/reports/conf/jdbcpds.conf`. If you have customized any of these files, you must apply the customizations to the corresponding files in the destination Oracle home.

> **Note:** To apply the customizations, you must copy the customized entries from the source Release 2 (9.0.2) instance to the destination 10*g* (9.0.4) instance. It is not safe to replace the 10*g* (9.0.4) file with the Release 2 (9.0.2) file, because the files are different.

In addition, you may want to perform the following optional manual steps:

- To preserve the cache files and the cache directory from Release 2 (9.0.2), copy the reports server cache directory from the source Oracle home to the destination Oracle home.
- To monitor additional reports server instances with the Oracle Process Manager and Notification Server (OPMN), define the reports server instances in `<destination_MT_OH>/opmn/conf/opmn.xml`.

  > **See Also:** Oracle Application Server Reports Services Publishing Reports to the Web, section entitled "Configuring Reports Server with Oracle Process Manager and Notification Server and Oracle Enterprise Manager"

- To manage additional reports server instances with the Oracle Enterprise Manager Application Server Control, define the reports server instances in `<destination_MT_OH>/sysman/emd/targets.xml`.

See Also: Oracle Application Server Reports Services Publishing Reports to the Web, section entitled "Configuring Reports Server with Oracle Process Manager and Notification Server and Oracle Enterprise Manager"

3.5.1 Upgrading User-Defined OC4J Instances With Oracle Application Server Reports Services Deployments

The OracleAS Upgrade Assistant upgrades the Oracle9iAS Release 2 (9.0.2) Business Intelligence & Forms configuration to the Oracle Application Server 10*g* (9.0.4) Business Intelligence & Forms configuration. It is not aware of OC4J instances outside of these configurations that may contain deployed reports, or of customizations made to those instances in order to enable the deployed reports to run.

Therefore, if you are using OC4J instances other than the standard Business Intelligence and Forms OC4J instance, you must apply any manual deployment steps that you performed on those instances in Oracle9iAS Release 2 (9.0.2) to the equivalent instances in Oracle Application Server 10*g* (9.0.4).

You must also add the Java option below to the java-options tag in *<destination_MT_OH>*/opmn/conf/opmn.xml to these OC4J instances before you can use them for Oracle Application Server Reports Services.

```
-Xbootclasspath^/p:<destination_MT_OH>/vbroker4/lib/vbjboot.jar
```

Example 3–4 Default java-options Tag

```
<data id="java-options" value="-Dhttp.proxyHost=www-proxy.us.oracle.com
-Dhttp.proxyPort=80"/>
```

Example 3–5 java-options Tag Modified for Oracle Application Server Reports Services

```
<data id="java-options" value="-Dhttp.proxyHost=www-proxy.us.oracle.com
-Dhttp.proxyPort=80"
-Xbootclasspath^/p:<destination_MT_OH>/vbroker4/lib/vbjboot.jar"/>
```

3.5.2 Integrating OracleAS Reports Services with Oracle Enterprise Manager

In the OracleAS 10*g* (9.0.4) reports server configuration file, the <identifier> element is encrypted, and matches the value in the *<destination_MT_OH>*/sysman/emd/targets.xml file.

During the upgrade, the OracleAS Upgrade Assistant copies the Oracle9*i*AS Release 2 (9.0.2) reports server configuration files into the OracleAS 10*g* (9.0.4) Reports configuration files. The <identifier> element in the Oracle9*i*AS Release 2 (9.0.2) is different from that in the 10*g* (9.0.4) targets.xml file, so some of the features of the OracleAS Reports Services will not work after the upgrade.

1. Edit the *<destination_MT_OH>*/reports/conf/*<server name>*.conf file and modify the <identifier> element to specify the user name and password and set the encrypted attribute to no. For example, if the user name is scott and the password is tiger:

   ```
   <identifier confidential="yes" encrypted="no>scott/tiger</identifier>
   ```

2. In the *<destination_MT_OH>*/sysman/emd/targets.xml file, locate the TYPE="oracle_repserv" and DISPLAY_NAME="Reports Server: *server name*" entries.

3. Set the `UserName` and `Password` properties to the same values as those in the `<identifier>` element in the `<server name>.conf` file. Set the `ENCRYPTED` attribute to `FALSE` for both properties.

> **Note:** The configuration changes will not be in effect until after the reports server and Oracle Enterprise Manager are both restarted.

3.5.3 Integrating Reports Services with OracleAS Portal

A Oracle9*i*AS Release 2 (9.0.2) Reports application can publish a report to Oracle9*i*AS Portal. By default, the Release 2 (9.0.2) Reports server configuration file contains a connection string that points to the Oracle9*i*AS Portal instance in the Infrastructure.

If you need to integrate OracleAS Reports Services with OracleAS Portal, you should upgrade to the OracleAS 10*g* (9.0.4) Business Intelligence and Forms installation type. If you upgrade to Forms and Reports Services in 10*g* (9.0.4), and still attempt to publish reports to a standalone or Infrastructure Portal instance, Oracle Corporation does not guarantee that that feature will function.

3.5.4 Integrating Reports Services with Oracle Internet Directory

A Oracle9*i*AS Release 2 (9.0.2) Reports application can store a connection string in Oracle Internet Directory in Oracle9*i*AS Release 2 (9.0.2) and execute a report using the `SSOCONN` parameter that points to the connection string in Oracle Internet Directory.

If you need to integrate OracleAS Reports Services with Oracle Internet Directory, you should upgrade to the OracleAS 10*g* (9.0.4) Business Intelligence and Forms installation type. If you upgrade to Forms and Reports Services in 10*g* (9.0.4), and still attempt to execute reports with the `SSOCONN` paramter, the report requests will fail. In this case, you must replace the `SSOCONN` parameter with a clear connection string.

3.6 Upgrading the tnsnames.ora File

The `<source_MT_OH>/network/admin/tnsnames.ora` file contains connection information for various databases. It is shared among Oracle Application Server components.

You should examine this file to determine whether it contains any entries that will be needed to operate a component in the 10*g* (9.0.4) installation, but which do not exist in the `<destination_MT_OH>/network/admin/tnsnames.ora` file. For example, entries might have been added since the Oracle9*i*AS Release 2 (9.0.2) installation to provide access to additional databases, or the entry for the Infrastructure Services repository might have been changed in some way.

If new or changed entries exist, you have two choices for upgrading: append individual entries, or copy the entire file.

3.6.1 Append New and Changed Entries to the 10*g* (9.0.4) tnsnames.ora File

Copy all new and changed entries to the `<destination_MT_OH>/network/admin/tnsnames.ora` file.

3.6.2 Copy the Oracle9*i*AS Release 2 (9.0.2) tnsnames.ora File to the 10*g* (9.0.4) Installation

If you are certain that doing so will not eliminate entries added or changed by other components, you can copy the entire `tnsnames.ora` file from the source to the destination Oracle home, restoring the `EXTPROC_CONNECTION_DATA.US.ORACLE.COM` entry (which is introduced in 10*g* (9.0.4)).

To upgrade by copying the entire file, follow the steps below:

1. Create a backup of `<source_MT_OH>/network/admin/tnsnames.ora` and `<destination_MT_OH>/network/admin/tnsnames.ora`.
2. Copy the entire `tnsnames.ora` file from the source to the destination Oracle home.
3. Copy the `EXTPROC_CONNECTION_DATA.US.ORACLE.COM` entry (used for executing external procedures in the database) from the 10*g* (9.0.4) backup file created in step 1 to the Oracle9*i*AS Release 2 (9.0.2) file.

3.7 Disabling Oracle Application Server Single Sign-On in the Upgraded Instance

Some applications upgraded from the source Oracle home may have been configured to use OracleAS Single Sign-On in the source Oracle home. Because the Forms and Reports Services installation type does not use OracleAS Single Sign-On, you must disable it in the destination Oracle home after the upgrade.

3.7.1 Disabling Oracle Application Server Single Sign-On in the Oracle HTTP Server

To disable Oracle Application Server Single Sign-On in the Oracle HTTP Server, you must comment out the Include directive for the `mod_osso.conf` file in the `<destination_MT_OH>/Apache/Apache/conf/httpd.conf` file, as shown below:

```
#include "<destination_MT_OH>/Apache/Apache/conf/mod_osso.conf"
```

3.7.2 Disabling Security Services in OracleAS Reports Services

By default, OracleAS Reports Services is enabled for OracleAS Single Sign-On. To disable it after the upgrade, you must edit the files that contain security-related configuration elements.

3.7.2.1 Editing the Server Configuration Files

You must edit the in-process server and reports server configuration files to eliminate security tags.

The in-process server configuration file is:

`<destination_MT_OH>/reports/conf/rep_<host name>.conf`

The reports server configuration file is:

`<destination_MT_OH>/reports/conf/<server name>.conf`

To disable OracleAS Single Sign-On in these files, you must:

- Comment out the security tags.
- Remove the `securityId` attribute from the job tag.

3.7.2.2 Editing the Servlet Properties File

The servlet properties file is:

`<destination_MT_OH>/reports/conf/rwservlet.properties`

You must set `SINGLESIGNON=NO` in this file.

3.7.3 Using OracleAS Forms Services Applications Without Single Sign-On

If OracleAS Forms Services was configured to run with OracleAS Single Sign-On in Oracle9*i*AS Release 2 (9.0.2), then the Single Sign-On configuration information will not be available in the OracleAS 10*g* (9.0.4) Forms and Reports Services installation after the upgrade. When OracleAS Forms Services applications are accessed after upgrade, users are prompted for login information via dialog box. The users will have to enter the information into the dialog at each access.

If there is a common user name and password for a given application, you may wish to provide the connect string in the application section of `<destination_MT_OH>forms90/server/formsweb.cfg` file, so that the login dialog box does not appear.

3.8 Port Values and the portlist.ini File After Upgrade

After you upgrade a middle tier to OracleAS 10*g* (9.0.4), the upgraded instance is using the same ports that the Oracle9iAS Release 2 (9.0.2) instance used. For this reason, after upgrade, you cannot start the source and destination middle tier instances at the same time, because of port conflicts.

The `<destination_MT_OH>/install/portlist.ini` file does not reflect the upgraded port settings; it lists the port values assigned by the installer before the upgrade. Table 3–2 lists pre- and post-upgrade values for Oracle HTTP Server, Oracle Enterprise Manager Application Server Control, and Oracle Application Server Web Cache.

Table 3–2 Port Values Before and After Upgrade

Component	Port in Source Oracle Home	Port Value in Destination Oracle Home Assigned by Installer and Recorded in portlist.ini File	Post-Upgrade Port Value
Oracle HTTP Server	Port: 7777 Listen: 7778	Port: 7783 Listen: 7784	Port: 7777 Listen: 7778
Oracle Enterprise Manager Application Server Control	1810	1812	1812
Oracle Application Server Web Cache	Administration: 4000 Invalidation: 4001 Statistics: 4002	Administration: 4003 Invalidation: 4004 Statistics: 4005	Administration: 4000 Invalidation: 4001 Statistics: 4002

3.9 Upgrading Application Server Clusters

Upgrading an application server cluster is a two-stage process. First, you must upgrade the configuration of one of the instances in the cluster from the source Oracle

home to the newly installed 10*g* (9.0.4) middle tier instance in the destination Oracle home. Then, you reconstruct the cluster by installing and clustering additional instances in new destination Oracle homes.

Follow these steps to upgrade an application server cluster:

1. Determine the instance name of the instance to upgrade with the command:

   ```
   <source_MT_OH>/dcm/bin/dcmctl listinstances
   ```

2. Stop all processes in the source instance with the command:

   ```
   <source_MT_OH>/dcm/bin/dcmctl shutdown
   ```

3. Stop all processes in the destination instance with the command:

   ```
   <destination_MT_OH>/dcm/bin/dcmctl shutdown
   ```

4. Follow the instructions in Section 1.3.3, "Performing an Upgrade with the OracleAS Upgrade Assistant (Graphical User Interface (GUI) Version)" on page 1-10 or Section 1.3.4, "Performing an Upgrade with the OracleAS Upgrade Assistant (Command-line Version)" on page 1-12.

5. Follow the instructions that apply to the upgraded configuration in the following sections:

   Section 3.1, "Completing the Oracle Application Server Containers for J2EE (OC4J) Upgrade" on page 3-1

   Section 3.2, "Completing the Oracle HTTP Server Upgrade" on page 3-8

   Section 3.3, "Completing the OracleAS Web Cache Upgrade" on page 3-10

   Section 3.4, "Completing the Oracle Application Server Forms Services Upgrade" on page 3-12

   Section 3.5, "Completing the Oracle Application Server Reports Services Upgrade" on page 3-13

   Section 3.6, "Upgrading the tnsnames.ora File" on page 3-15

   Section 3.7, "Disabling Oracle Application Server Single Sign-On in the Upgraded Instance" on page 3-16

   Section 3.9, "Upgrading Application Server Clusters" on page 3-17

6. Start the Oracle Enterprise Manager Application Server Control with the command:

   ```
   <destination_MT_OH>/bin/emctl start iasconsole
   ```

7. Display the OracleAS Instance Home Page.

8. Start the upgraded instance by following the instructions in Section 3.10, "Starting the Upgraded Forms and Reports Services Instance" on page 3-19.

9. Create a cluster with the command:

   ```
   <destination_MT_OH>/dcm/bin/dcmctl createcluster <name of cluster>
   ```

10. Join the upgraded instance to the cluster with the command:

    ```
    <destination_MT_OH>/dcm/bin/dcmctl joincluster -cl <name of cluster>
    ```

11. Join each additional instance to the cluster with the command:

```
<destination_MT_OH>/dcm/bin/dcmctl joincluster -cl <name of cluster>
```

12. If the upgraded configuration used `mod_oc4j.conf` for request routing, do the following:

    a. View the `<destination_MT_OH>/Apache/Apache/conf/mod_oc4j.conf` file in one of the instances, noting the instance and cluster names in the `Oc4jMount` directives.

    b. Change the instance (and, if necessary) cluster names to the instance name of the upgraded instance.

    c. Copy the `Oc4jMount` directives to the `mod_oc4.conf` file in each instance in the new cluster.

    d. Verify that requests that match the URL patterns in the `Oc4jMount` directives are routed to the correct instances.

3.10 Starting the Upgraded Forms and Reports Services Instance

After the OracleAS Upgrade Assistant has finished processing, and you have completed all of the applicable manual upgrade tasks, start the upgraded Forms and Reports Services instance.

Follow these instructions to start the middle tier instance:

1. Start the application server components by issuing this command:

   ```
   <destination_MT_OH>/opmn/bin/opmnctl startall
   ```

   Oracle Process Management and Notification and all of the processes it manages are started (i.e., Distributed Configuration Management, Oracle HTTP Server, OC4J instances, OracleAS Web Cache, Oracle Application Server Forms Services, and Oracle Application Server Reports Services).

2. Start the Application Server Control by issuing this command:

   ```
   <destination_MT_OH>/bin/emctl start iasconsole
   ```

3.10.1 Resolving Oracle Call Interface Component Errors

If the upgrade succeeds, but components that use the Oracle Call Interface do not function properly, it may be that the `tnsnames.ora` file was not upgraded, or was upgraded incorrectly. Review the instructions and upgrade strategies presented in Section 3.6, "Upgrading the tnsnames.ora File" on page 3-15 and verify that the file contains all necessary entries for the components that use it.

3.11 Validating the Forms and Reports Services Upgrade

This section describes tasks you should perform after the upgrade to validate that the upgrade was successful.

3.11.1 Verify Operation of Middle Tier Components

Follow these steps to verify that the middle tier components that were upgraded are started:

1. In a browser, access the Oracle Enterprise Manager Application Server Control in the 10*g* (9.0.4) middle tier Oracle home by entering its URL. Ensure that you provide the correct host name and port number. For example:

```
http://midtierhost.mycompany.com:1812
```

The Oracle Enterprise Manager page appears. A link for the middle tier instance appears in the **Standalone Instances** section.

2. Click the link.

   The **System Components** page appears.

3. Verify that the components are running.

4. Verify that the configuration information for the components in use is reflected in the 10*g* (9.0.4) Oracle home.

3.11.2 Check Significant URLs

Follow these steps to verify that you can access the Oracle HTTP Server and application URLs:

1. Verify that you can access the Oracle HTTP Server on the same host and port that you did in the previous release by entering the URL. Ensure that you provide the correct host name and port number. For example:

   ```
   http://midtierhost.mycompany.com:7777
   ```

2. Verify that you can access the URLs for the applications you operated in the previous release, and ensure that the applications are functioning as they did in the previous release.

3.12 Considerations for the Source Oracle Home After Upgrade

The upgrade process leaves the source Oracle home unchanged. Depending on the type of installation you have, and your future needs, you may elect to remove the source Oracle home, or to retain it for specific reasons. This section discusses decommissioning the source Oracle home, as well as reasons for retaining it, including the steps you must perform to revert to using it.

Note: If you retain the source Oracle home, you cannot operate it simultaneously with the destination Oracle home, because certain components have the same port values after upgraded. See Section 3.8, "Port Values and the portlist.ini File After Upgrade" on page 3-17.

3.12.1 Decommissioning the Source Oracle Home

When you are certain that the upgrade was successful, you have all of the necessary backups, and have no plans to revert to the source Oracle home, you may elect to remove the files from the source Oracle home. Use the Oracle Universal Installer to deinstall the instance.

> **See Also:** *Oracle9i Application Server Installation Guide* in the Release 2 (9.0.2) documentation library

Deinstalling a Release 2 (9.0.2) or instance when there is also a 10*g* (9.0.4) instance on the computer requires a patch, and there are issues associated with this deinstallation that may apply to your configuration.

> **See Also:** *Oracle Application Server 10g Installation Guide*

3.12.1.1 Preserving Application Files and Log Files

If there are application files or log files in the source Oracle home that are being referenced or used by the destination Oracle home, you should move them to another location before you decommission the source Oracle home, and, in the destination Oracle home, change any references to the files to the new location.

Index

Symbols

A

C

D

E

F

H

I

J

L

M

O

P

Q

R

S

T

U

V

W

X