Oracle® Universal Installer

and OPatch User's Guide

11*g* Release 2 (11.2) for Windows and UNIX

E12255-11

March 2011

ORACLE®

Oracle Universal Installer and OPatch User's Guide, 11*g* Release 2 (11.2) for Windows and UNIX

E12255-11

Primary Author: Michael Zampiceni

Contributor: Smitha Viswanathan, Sumant Sankaran, Phi Nguyen

Contents

2   Managing Oracle Homes

3   Customizing and Creating Response Files

4   Installing Products

5   Installing Cluster Environments

8   Oracle Internationalization and Translation

A   Frequently Asked Questions

B   Troubleshooting and Debugging Oracle Universal Installer

C   Understanding Oracle Universal Installer Commands

D   Sample Files

E   Error Messages and Solutions

Index

Preface

This manual describes how to manage Oracle and third-party software using the Oracle Universal Installer and OPatch. This guide provides instructions for installing, setting up, and starting the Oracle Universal Installer software. It also provides instructions on cloning and patching Oracle software.

Audience

This manual is intended for users installing Oracle software products using the Oracle Universal Installer and covers only the generic functionality and concepts. Use this manual in conjunction with any product-specific installation guides available with your Oracle product.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711.

Conventions

The following text conventions are used in this document:

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

Related Documents

For additional information, see the Oracle Fusion Middleware Patching Guide.

1
Introduction to Oracle Universal Installer

This chapter is designed to aid developers, administrators, and all other users who install Oracle software by understanding the system requirements, features, and key concepts of Oracle Universal Installer.

This chapter includes the following sections:

- System Requirements
- Key Features
- Utilities
- Actions Performed by the Utilities
- Major Entities Created
- Modes of Installation

System Requirements

The minimum system requirements for Oracle Universal Installer are as follows:

- **Java Runtime Environment (JRE)** — Automatically installed with Oracle Universal Installer on most platforms. Check the Release Notes or installation guide of the products that you are installing for the required version.

- **Memory Requirements** — Memory requirements vary depending on the number of components installed. Check the Release Notes or installation guide for the products that you are installing for details. 32 MB is the minimum recommended on all platforms.

- **Disk Space Requirements** — Oracle recommends at least 200 MB for Oracle Universal Installer files on Windows platforms and 116 MB on UNIX. (UNIX requires more memory because of the difference in JRE sizes for the platforms.) You may need up to 1 MB for the **related** inventory files.

  When you run Oracle Universal Installer from an NFS-mounted user home, especially for Linux, execute the quota command to check the space availability. Never perform an installation on a user home for which space is allocated based on quota.

Key Features

Oracle Universal Installer 11*g* Release 2 (11.2) offers the following features:

- **An XML-based centralized inventory**

The XML format enables third-party Java applications to query the inventory for information about installed software.

- **Cloning of existing Oracle homes**

  Enables you to copy an existing Oracle home to another location and "fix it up" by updating the installation configuration to be specific to the new environment. Cloning makes it easy to propagate a standard setup without having to install and configure after installation.

- **Better support for cluster environments**

  Oracle Universal Installer now replicates its inventory to all nodes that participate in a cluster-based installation. You can invoke Oracle Universal Installer from any node on the cluster that is part of the installation. You can then upgrade, remove, or patch existing software from any node.

- **True silent capability**

  When running Oracle Universal Installer in silent mode on a character mode console, you no longer need to specify an X-server or set the DISPLAY environment variable on UNIX. No GUI classes are instantiated, making the silent mode truly silent.

- **Ability to record your Oracle Universal Installer session to a response file**

  This feature makes it easy to duplicate the results of a successful installation on multiple systems. All the options you selected during the installation are saved in the resulting response file.

- **More accurate disk space calculations**

  Oracle Universal Installer now uses a more accurate method of calculating the disk space your Oracle products require. This feature reduces the risk of running out of disk space during an installation.

- **Automatically launched software after installation**

  Some Oracle products now take advantage of a new feature that enables the software to launch automatically immediately after the installation.

- **Cleaner deinstallation and upgrades**

  Deinstallation completely removes all software, leaving no "bits" behind. This also completely removes files associated with configuration assistants and patchsets. Oracle homes can also be removed from the inventory and registry. For deinstalling 11.2 Oracle Clusterware, Database, and client homes, OUI prompts you to run the deinstall/deconfig utility from the home.

- **Integrated prerequisite checking**

  Provides a prerequisite checking tool to diagnose the readiness of an environment for installation. The prerequisite checks are run as part of the installation process, but can also be run as a separate application.

- **Support for Desktop Class and Server Class**

  The following installation types are available for the database:

  – **Desktop Class**

    Choose this option if you are installing on a laptop or desktop class system. This option includes a starter database and provides minimal configuration. This option is designed for users that want to quickly bring up and run the database.

–   **Server Class**

Choose this option if you are installing on a server class system, such as what you would use when deploying Oracle in a production data center. This option provides more advanced configuration options. Advanced configuration options available using this installation type include Oracle RAC, Automatic Storage Management, backup and recovery configuration, integration with Enterprise Manager Grid Control, and more fine-grained memory tuning, as well as other options.

For the Server Class option, the Typical Installation method is selected by default. It enables you to quickly install the Oracle Database using minimal input. This method installs the software and optionally creates a general-purpose database using the information that you specify in this dialog.

Utilities

Oracle offers two utilities for software deployment:

■   Oracle Universal Installer to install Oracle products

■   OPatch to apply interim patches

Oracle Universal Installer

Oracle Universal Installer 11*g* Release 2 (11.2) is a Java-based installer that enables you to install Oracle components from CDs or from a staged HTTP location. It performs component-based installations as well as complex installations, such as integrated bundle and suite installations, and installations over the Web.

OPatch

OPatch is an Oracle-supplied utility that assists you with the process of applying interim patches to Oracle's software. OPatch 11.2 is a Java-based utility that can run on either OUI-based Oracle homes or standalone homes. It works on all operating systems for which Oracle releases software. For more information on OPatch, see the *Oracle OPatch User's Guide*.

Actions Performed by the Utilities

Oracle Universal Installer and OPatch performs the following activities:

Installation

Installation is the process of choosing products from a release or stage area and deploying them in the target directory. There are four modes of installation, which are explained in the section "Modes of Installation" on page 1-5.

Deinstallation

Deinstallation is the process of removing an installed product from the installation area. A de-installation can be cancelled, resumed or rolled back. It can be executed either interactively or silently.

Cloning

Cloning is the process of copying an existing installation to a different location while preserving its configuration. You can install multiple copies of the Oracle product easily on different computers using cloning. During cloning, Oracle Universal Installer is invoked in clone mode to adapt the home to the target environment. Oracle Universal Installer in clone mode replays all the actions that have been executed to originally install the Oracle home. The difference between installation and cloning is that during cloning, Oracle Universal Installer runs the actions in the clone mode. Each action decides how to respond during cloning.

Patching

During patching, a small collection of files are copied over an existing installation to fix certain bugs. OPatch is an Oracle-supplied utility that facilitates Oracle software patching. For more information on OPatch, see the *Oracle OPatch User's Guide*.

Upgrades and Patchsets

Oracle Universal Installer enables you to upgrade a product from one version to another version. An upgrade is a major product enhancement that often requires installation of the upgraded software. For example, you may want to convert your Oracle Database 10*g*R2 (10.2) to Oracle 11*g*R2 (11.2) Database, which is called an upgrade.

A group of patches form a patchset. For example, you may want to convert your Oracle 11*g*R1 (11.1) Database from version 11.1 to version 11.1.0.7.0, which is called applying a patchset.

Adding Nodes

You can install an Oracle home on multiple nodes in a cluster. You can extend the cluster for a particular Oracle home using the `-addNode` flag of Oracle Universal Installer. You can add more than one node to the Oracle home. The Oracle Universal Installer with the `-addNode` flag is always run on the local node and not on the node(s) to be added. You can add nodes to an Oracle Clusterware node or an Oracle Real Application Clusters node depending upon whether the node addition is being performed at the Oracle Clusterware layer or the Oracle Real Application Clusters database layer. You can use `$OH/oui/bin/addNode.sh` to add nodes.

For more information on adding nodes, see "Installing Cluster Environments" on page 5-1.

Inventory Operations

Oracle Universal Installer supports the following inventory operations:

Attaching Oracle Homes

Oracle Universal Installer uses the `-attachHome` flag to attach an Oracle home to the inventory to set up the Central Inventory or to register an existing Oracle home with the Central Inventory. You can use `attachHome.sh` (bat) from an Oracle home as well as from the shiphome.

For more information, see "Creating the Central Inventory" on page 2-8.

Detaching Oracle Homes

Oracle Universal Installer uses the `-detachHome` flag to remove an Oracle home from the Central Inventory. You can use `detachHome.sh` (bat) from an Oracle home.

For more information, see "Detaching Oracle Homes from the Central Inventory" on page 2-9.

Updating the Node List

Oracle Universal Installer uses the `-updateNodeList` flag to update the node list in the inventory. For an Oracle Clusterware home, you need to pass CRS=true from the updateNodeList command line.

For more information, see "Updating the Nodes of a Cluster" on page 2-15.

Major Entities Created

The following major entities are created when you run Oracle Universal Installer.

Inventory

The Oracle Universal Installer inventory stores information about all Oracle software products installed in all the Oracle homes on a host, provided the product was installed using Oracle Universal Installer. The inventory is organized as follows:

- Central Inventory
- Oracle home inventory

For more information on the inventory and the structure of the inventory, see "Oracle Universal Installer Inventory" on page 2-4.

Oracle Home

An Oracle home is the system context in which the Oracle products run. This context consists of the following:

- Directory location where the products are installed
- Corresponding system path setup
- Program groups associated with the products installed in that home (where applicable)
- Services running from that home

For more information on Oracle homes, see Chapter 2, "Managing Oracle Homes".

Modes of Installation

You can run the Oracle Universal Installer in the following modes:

Interactive Mode

You can use the interactive mode to walk through the installation by providing information in the dialogs when prompted. This method is useful when installing a small number of products in different setups on a small number of hosts.

Suppressed-Interactive Mode

You can use this mode to supply the necessary information by using a combination of a response file or command line entries with certain interactive dialogs. This is useful when an installation has a common set of parameters that can be captured in a response file, in addition to the custom information that you must enter manually.

Silent Mode

You can use this mode to bypass the Graphical User Interface (GUI) and supply the necessary information in a response file. This method is useful when installing the same product multiple times on multiple hosts. By using the response files, you can also automate the installation of a product for which you know the installation parameters.

For more information on silent installation, see Chapter 3, "Customizing and Creating Response Files".

Cluster Mode

A cluster installation uses Oracle Universal Installer to install software on the nodes of a cluster that are network-reachable and bound together by Oracle Clusterware. You can use Oracle Universal Installer to extend the Oracle home of a product installation to include additional nodes on the cluster. You need to install Oracle Clusterware for a cluster installation.

For more information on cluster installations, see Chapter 5, "Installing Cluster Environments".

2

Managing Oracle Homes

This chapter contains the following sections:

- Introduction to Oracle Homes
- Introduction to Oracle Base
- Installing an Oracle Product
- Removing Oracle Homes
- Determining the Default Oracle Home
- Multiple Oracle Homes
- Oracle Universal Installer Inventory
- Home Selector (Available on Win32 Platforms)

Introduction to Oracle Homes

The Oracle Universal Installer supports the installation of several active Oracle homes on the same host. An Oracle home is a directory into which all Oracle software is installed. This is pointed to by an environment variable. The Oracle home consists of the following:

- Directory location where the products are installed
- Corresponding system path setup
- Program groups associated with the products installed in the home (where applicable)
- Services running from the home

Introduction to Oracle Base

The Oracle base location is the location where Oracle Database binaries are stored. During installation, you are prompted for the Oracle base path. Typically, an Oracle base path for the database is created during Oracle Grid Infrastructure installation.

To prepare for installation, Oracle recommends that you only set the ORACLE_BASE environment variable to define paths for Oracle binaries and configuration files. Oracle Universal Installer (OUI) creates other necessary paths and environment variables in accordance with the Optimal Flexible Architecture (OFA) rules for well-structured Oracle software environments.

For example, with Oracle Database 11g, Oracle recommends that you do not set an Oracle home environment variable allow OUI to create it instead. If the Oracle base

path is /u01/app/oracle, then by default, OUI creates the following Oracle home path:

```
/u01/app/oracle/product/11.2.0/dbhome_1
```

Ensure that the paths you select for Oracle software, such as Oracle home paths and the Oracle base path, use only ASCII characters. Because installation owner names are used by default for some paths, this ASCII character restriction applies to user names, file names, and directory names.

Installing an Oracle Product

When you install an Oracle product, an Oracle home is created.

To install the product and create the Oracle home, perform the following steps:

1. Run Oracle Universal Installer.

2. In the Specify Home Details page, enter the Oracle home settings for the installation session. See Table 2–1 for a description of the fields in this section of the screen.

3. Continue with your installation. See Chapter 4, "Installing Products" for detailed information.

Table 2–1   Oracle Installation Settings for Specify Home Details page

Settings	Functions
Name	Enter a name for the Oracle home. This name identifies the program group associated with a particular home and the Oracle services installed on this home. The Oracle home name must be between 1 to 127 characters long, and can include only alphanumeric characters and underscores.
Software Location	Enter the full path to an Oracle home, or select an Oracle home from the drop-down list of existing Oracle homes. The Oracle home location is the directory where products are installed.
	Data files may or may not be installed within an Oracle home. You can use the **Browse** button to choose a directory to install your product.
	For Windows platforms, you must provide a valid path that is not in the Windows directory. Different homes cannot share the same location.

> **Note:**   Oracle recommends that you designate an Oracle home location that is an empty or non-existing directory. If you select a directory for the Oracle home location that is not empty or already exists, you will be warned and asked if you want to proceed.
>
> For silent installations, if a non-empty, existing directory is specified, an error is logged in the console and in the `installActions<timestamp>.log` file. Also, the Oracle Universal Installer aborts. To override this condition, use the `-force` flag on the command line. The effect of using the `-force` flag is the same as selecting **Yes** while installing in interactive mode. You receive a warning message, but the installation continues.

Removing Oracle Homes

To remove or deinstall Oracle homes, you can either use the Deinstall tool included with the Shiphome, or use the Deinstall utility available as part of the Oracle home.

- To use the Deinstall tool, do the following to remove the Oracle home:

    1. cd to DeinstallTool.

    2. Run deinstall -home <OracleHome>.

- To use the Deinstall utility, do the following to remove the Oracle home:

    1. cd to $Oracle_Home/deinstall.

    2. Run Deinstall.

Determining the Default Oracle Home

By default, when you start Oracle Universal Installer, the software searches your system to determine the default Oracle home where Oracle software should be installed.

In all cases, the ORACLE_HOME name is taken first from the command line if it is specified, or else from the response file if specified. If not specified, the value of DEFAULT_ORACLE_HOME_NAME in oraparam.ini is examined. Typically, the following convention is used for the name:

Ora<*short_marketing_name_in_oraparam.ini>_<home_identifier*>

Where <*short_marketing_name_in oraparam.ini*> is the short product marketing name as specified in the oraparam.ini file; for example, "Db11g", and <*home_identifier*> is a counter derived from the Central Inventory. For example, the ORACLE_HOME name could be OraDb11g_1.

The ORACLE_HOME **path** is taken first from the command line if specified, or else from the response file if specified. If not, the ORACLE_HOME environment variable is used. If neither is specified, the following conventions are used for the path:

- If ORACLE_BASE *has* been specified in the environment:

    $ORACLE_BASE/product/<*version>/<short_name>_<counter*>

    For example: $ORACLE_BASE/product/11.2.0/Db_1.

- If ORACLE_BASE *has not* been specified in the environment:

    $HOME/product/<*version>/<short_name>_<counter*>

    Where <*short_name*> is the short product name; for example, "Db", and <*counter*> is picked up based on the existence of the files. For example, the ORACLE_HOME path could be $HOME/product/11.2.0/Db_1.

The instance-related directory location is accepted first from the response file, if specified. If not, the oradata, flash_recovery_area, admin, and doc directories are created under ORACLE_BASE. If ORACLE_BASE has not been specified, the default is the $ORACLE_HOME/oradata directory. If the parent directory of the Oracle home is writable, these directories are created in the parent directory of the Oracle home.

Multiple Oracle Homes

Oracle Universal Installer supports the installation of several active Oracle homes on the same host as long as the products support this at run-time. Multiple versions of the same product or different products can run from different Oracle homes concurrently. Products installed in one home do not conflict or interact with products installed on another home. You can update software in any home at any time, assuming all Oracle applications, services, and processes installed on the target home are shut down. Processes from other homes may still be running.

Target Home

The Oracle home currently accessed by Oracle Universal Installer for installation or deinstallation is the target home. To upgrade or remove products from the target homes, these products must be shut down or stopped.

Oracle Universal Installer Inventory

The Oracle Universal Installer inventory stores information about all Oracle software products installed in all Oracle homes on a host, provided the product was installed using Oracle Universal Installer.

Inventory information is stored in Extensible Markup Language (XML) format. The XML format enables easier diagnosis of problems and faster loading of data. Any secure information is not stored directly in the inventory. As a result, during removal of some products, you may be prompted to enter the required credentials for validation.

> **Note:** Oracle recommends placing the central inventory on a local file system that is not shared by other systems, since the central inventory is a system-specific inventory of the installations on this system. It is strongly recommended that you place the central inventory on a local disk so that installations from other systems do not corrupt the inventory. You should not place the central inventory in the Oracle Base.

Structure of the Oracle Universal Installer Inventory

The Oracle Universal Installer inventory has the following hierarchical structure:

- Central Inventory Pointer File
- Central Inventory
- Oracle Home Inventory

Central Inventory Pointer File

Every Oracle software installation has an associated Central Inventory where the details of all the Oracle products installed on a host are registered. The Central Inventory is located in the directory that the inventory pointer file specifies. Each Oracle software installation has its own Central Inventory pointer file that is unknown to another Oracle software installation.

For Oracle homes sharing the same Central Inventory, the Oracle Universal Installer components perform all read and write operations on the inventory. The operations on the Central Inventory are performed through a locking mechanism. This implies that

when an operation such as installation, upgrade, or patching occurs on an Oracle home, these operations become blocked on other Oracle homes that share the same Central Inventory.

Table 2–2 shows the location of the default inventory pointer file for various platforms:

Table 2–2   Default Inventory Pointer File Locations

Platform	Default Inventory Pointer Location
Linux Linux.PPC64 AIX	/etc/oraInst.loc
Solaris.SPARC Solaris.X64 HPUX HPIA HP.TRU64 Linux.IA64 Linux.xSeries	/var/opt/oracle/oraInst.loc
Windows Windows.X64 Windows.IA64	HKEY_LOCAL_MACHINE/Software/ Oracle/inst.loc

The following string shows an example of the path for the `oraInst.loc` file:

```
inventory_loc=/home/oracle_db11g/product/11.2.0/db_1
inst_group=oracle
```

In UNIX, if you do not want to use the Central Inventory located in the directory specified by the inventory pointer file, you can use the `-invPtrLoc` flag to specify another inventory pointer file. The syntax is as follows:

```
./runInstaller -silent -invPtrLoc <Location_of_oraInst.loc> ORACLE_
HOME="<Location_of_Oracle_Home>"
```

> **Note:**   If the contents of the `oraInst.loc` file is empty, Oracle Universal Installer prompts you to create a new inventory.

Central Inventory

The Central Inventory contains the information relating to all Oracle products installed on a host. It contains the following files and folders:

- Inventory File

- Logs Directory

Inventory File   This file lists all the Oracle homes installed on the node. For each Oracle home, it also lists the Oracle home name, home index, and nodes on which the home is installed. It also mentions if the home is an Oracle Clusterware home or a removed Oracle home. It can only detect removed Oracle homes created using Oracle Universal Installer version 11.2 and later. This file is present in the following location:

```
<central inventory location>/ContentsXML/inventory.xml
```

The following code shows a sample `inventory.xml` file:

```
<?xml version="1.0" standalone="yes" ?>
```

```
<!-- Copyright (c) 2009 Oracle Corporation. All rights Reserved -->
<!-- Do not modify the contents of this file by hand. -->
<INVENTORY>
<VERSION_INFO>
   <SAVED_WITH>11.2.0.0.0</SAVED_WITH>
   <MINIMUM_VER>2.1.0.6.0</MINIMUM_VER>
</VERSION_INFO>
<HOME_LIST>
<HOME NAME="OraDb11g_home1" LOC="/home/oracle_db11g/product/11.2.0/db_1" TYPE="O"
IDX="1"/>
<HOME NAME="OUIHome" LOC="D:\OraHome1" TYPE="O" IDX="2" />
<HOME NAME="OUIHome12" LOC="D:\OraHome12" TYPE="O" IDX="3" />
<HOME NAME="OUIHome1" LOC="D:\homes\OraHome1" TYPE="O" IDX="4" />
</HOME_LIST>
</INVENTORY>
```

> **Note:** Oracle recommends that you do not remove or manually edit this file as it could affect installation and patching.

Logs Directory   The Central Inventory contains installation logs in the following location:

```
<central inventory location>/logs
```

The logs directory contains the logs corresponding to all installations performed on a particular node. You can also find a copy of the installation log in the $ORACLE_HOME/cfgtoollogs directory.

The installation logs for an installation are identified by the timestamp associated with the log files. These files are generally saved in the following format:

```
<Name_of_Action><YYYY-MM-DD_HH-MM-SS{AM/PM}>.log
```

For example, consider an attachHome operation performed on 17th, May, 2009 at 6.45AM. The associated log file would be created as follows:

```
AttachHome2009-05-17_06-45-00AM.log
```

Oracle Home Inventory

Oracle home inventory or local inventory is present inside each Oracle home. It only contains information relevant to a particular Oracle home. This file is located in the following location:

```
$ORACLE_HOME/inventory
```

It contains the following files and folders:

- Components File
- Home Properties File
- Other Folders

Components File   This file contains the details about third-party applications like Java Runtime Environment (JRE) required by different Java-based Oracle tools and components. In addition, it also contains details of all the components as well as patchsets or interim patches installed in the Oracle home. This file is located here:

```
ORACLE_HOME/inventory/ContentsXML/comps.xml
```

For an example of the components file, see "Sample Components File" on page D-10.

Home Properties File   This file contains the details about the node list, the local node name, and the `Oracle Clusterware` flag for the Oracle home. In a shared Oracle home, the local node information is not present. This file also contains the following information:

- **GUID —** Unique global ID for the Oracle home

- **ARU ID** — Unique platform ID. The patching and patchset application depends on this ID.

- **ARU ID DESCRIPTION** — Platform description

The information in `oraclehomeproperties.xml` overrides the information in `inventory.xml`. This file is located here:

```
$ORACLE_HOME/inventory/ContentsXML/oraclehomeproperties.xml
```

The following example shows the Oracle home property file:

```
<GUID>893051798#.356758136</GUID>
    <HOME CRS="T"/>
    <ARU_PLATFORM_INFO>
        <ARU_ID>46</ARU_ID>
        <ARU_ID_DESCRIPTION>Linux x86</ARU_ID_DESCRIPTION>
    </ARU_PLATFORM_INFO>
    <CLUSTER_INFO>
        <LOCAL_NODE NAME="stacg30"/>
     <NODE_LIST><NODE NAME="stacg34"/><NODE
NAME="stacg30"/></NODE_LIST></CLUSTER_INFO>
```

Table 2–3 lists the ARU_IDs for some platforms:

Table 2–3   ARU_IDs for Platforms

Platform	ARU_ID
Apple Mac OS X (PowerPC)	421
HP Tru64 UNIX	87
HP-UX Itanium	197
HP-UX Itanium (32-bit)	278
HP-UX PA-RISC (32-bit)	2
HP-UX PA-RISC (64-bit)	59
IBM AIX on POWER Systems (32-bit)	319
IBM AIX on POWER Systems (64-bit)	212
IBM i on POWER Systems	43
IBM S/390-based Linux (31-bit)	211
IBM z/OS on System z	30
IBM: Linux on POWER Systems	227
IBM: Linux on System z	209
Linux Itanium	214
Linux x86	46
Linux x86-64	226

Table 2–3   (Cont.) ARU_IDs for Platforms

Platform	ARU_ID
Microsoft Windows (32-bit)	912
Microsoft Windows Itanium (64-bit)	208
Microsoft Windows x64 (64-bit)	233
Sun Solaris SPARC (32-bit)	453
Sun Solaris SPARC (64-bit)	23
Sun Solaris x86 (32-bit)	173
Sun Solaris x86-64 (64-bit)	267

Other Folders   Table 2–4 lists the other folders you can find in the Oracle home inventory:

Table 2–4   Other Folders in the Oracle Home Inventory

Folder Name	Description
Scripts	Contains the scripts used for the cloning operation.
ContentsXML	Contains the details of the components and libraries installed.
Templates	Contains the template files used for cloning.
oneoffs	Contains the details of the one-off patches applied.

Creating the Central Inventory

Oracle Universal Installer enables you to set up the Central Inventory on a clean host or register an existing Oracle home with the Central Inventory when it is lost or corrupted. If the Central Inventory does not already exist, Oracle Universal Installer creates the Central Inventory in the location specified by the `oraInst.loc` file.

You can set up the Central Inventory by using the `-attachHome` flag of Oracle Universal Installer. The syntax is as follows:

```
./runInstaller -silent -attachHome ORACLE_HOME="<Oracle_Home_Location>"
"CLUSTER_NODES={<node1,node2>}" LOCAL_NODE="<node_name>"
```

Using the Session Variables

You can use the following session variables:

- `ORACLE_HOME`
- `CLUSTER_NODES`
- `REMOTE_NODES`
- `LOCAL_NODE`

In a cluster installation:

- If you do not pass the `CLUSTER_NODES` session variable, Oracle Universal Installer takes it from the `$ORACLE_HOME/inventory/ContentsXML/oraclehomeproperties.xml` file.

- If you do not pass the `LOCAL_NODE` session variable, Oracle Universal Installer takes it from the `$ORACLE_HOME/inventory/ContentsXML/oraclehomeproperties.xml`

file. If it does not find an entry there, it takes it from the Oracle Clusterware stack. If it is not able to find it in the stack, the first node of the CLUSTER_NODES is taken as the LOCAL_NODE.

You can pass the REMOTE_NODES variable if you want to specify the list of remote nodes. If you want to set up the Central Inventory in the local node, you need to pass the -local flag, and the REMOTE_NODES variable is empty. The syntax is as follows:

```
./runInstaller -silent -attachHome -invPtrLoc ./oraInst.loc
ORACLE_HOME="<Oracle_Home_Location>" "REMOTE_NODES={}" -local
```

> **Note:** When you use the -local flag, it performs the action on the local node irrespective of the cluster nodes specified.

For a non-Oracle Real Application Clusters setup, you do not need to pass the LOCAL_ NODE variable, and the CLUSTER_NODES variable is empty. The syntax is as follows:

```
./runInstaller -silent -attachHome -invPtrLoc ./oraInst.loc
ORACLE_HOME="<Oracle_Home_Location>" "CLUSTER_NODES={}"
```

You can use the -local flag to attach the local Oracle home. If you are using a shared Oracle home with the -local flag, use the -cfs flag. This ensures that the local node information is not populated inside a shared Oracle home.

Verifying the Operation

After attaching the Oracle home, you can verify the success of the operation by verifying the contents of the log file present in the <central_inventory>/logs directory. You can also view the contents of the inventory.xml file under the <central-inventory>/ContentsXML directory to verify if the Oracle home is registered.

> **Note:** Oracle recommends cloning on Windows operating systems to create the Central Inventory.

Detaching Oracle Homes from the Central Inventory

You can detach an Oracle home from the Central Inventory. When you pass this flag, it updates the inventory.xml file present in the Central Inventory. The syntax is as follows:

```
./runInstaller -silent -detachHome -invPtrLoc ./oraInst.loc
ORACLE_HOME="<Oracle_Home_Location>"
```

Using the Optional -local Flag

You can use the -local flag to detach the Oracle home from the inventory of the local node. If you are using a shared Oracle home, use the -cfs flag. This ensures that the local node information is not populated inside a shared Oracle home.

```
./runInstaller -silent -local -cfs -detachHome
ORACLE_HOME="<Oracle_Home_Location>"
```

Removing the Central Inventory

Even after all the Oracle homes on a host are removed, you will find traces of the inventory with certain log files. If you do not want to maintain these files and want to remove the Central Inventory, do the following:

Removing the Central Inventory on UNIX Platforms   You can remove the Central Inventory on UNIX by performing the following steps:

1. Locate the `oraInst.loc` file and get the Central Inventory location (`inventory_loc` parameter) from this file.

   For Solaris, this file is located in the `/var/opt/oracle` folder.

   For Linux, this file is located in the `/etc` folder.

2. Remove the Central Inventory by executing the following command:

   `rm -rf <central_inventory_location>`

3. Remove the `oraInst.loc` file by executing the following command with root privileges:

   Solaris:

   `rm /var/opt/oracle/oraInst.loc`

   Linux:

   `rm /etc/oraInst.loc`

Removing the Central Inventory on Windows Platforms   You can remove the Central Inventory on Windows by performing the following steps:

1. Locate the registry key:

   `\\HKEY_LOCAL_MACHINE\Software\Oracle\inst_loc`

2. Get the Central Inventory location from this key.

3. Delete the Central Inventory directory and all its contents.

4. Delete the registry key:

   `\\HKEY_LOCAL_MACHINE\Software\Oracle\inst_loc`

Consolidating Multiple Central Inventories

The following procedure explains how to consolidate multiple central inventories into a single central inventory. For Oracle Clusterware and Oracle RAC homes, perform this procedure for each node.

1. Identify the Central Inventory to use and ensure that it is the same path on all nodes of the cluster.

2. Go to this Central Inventory directory and run orainstRoot.sh to ensure that the oraInst.loc file points to this inventory.

3. Identify the other central inventories on the system, then identify the Oracle homes for each Central Inventory.

4. Do the following for each Oracle home:

   a. Enter cd $ORACLE_HOME/oui/bin.

b. Run ./attachhome.sh -silent -local "CLUSTER_NODES=
{<List of nodes>}" LOCAL_NODE=<nodename> CRS="TRUE"

For single instance homes, run ./attachhome.sh -silent.

c. Verify the inventory updates by going to the ORACLE_HOME/OPatch directory and running opatch lsinventory -detail.

d. Verify that the overall inventory is being updated by running ./runInstaller and clicking on 'Installed Products...' to bring up the contents of the central inventory. Ensure that the inventory shows the new home and the nodes.

Reconstructing a Shared Central Inventory on a Non-shared Location

The following sections provide advisory information about shared Central Inventories, explain the enforced shared inventory check, and provide a procedure for reconstructing the Central Inventory.

Oracle Recommendation Regarding Central Inventory Placement

The Oracle Central Inventory is a repository for all Oracle products (software) installed on a system. Since the Central Inventory consists of system-specific information, it is required that the Central Inventory be saved on a local non-shared directory on the system. While software can be shared across nodes, the inventory should be local to each system.

If the Central Inventory is shared across system (nodes), this would reflect the installation information for a group of systems collectively. Such a configuration also mandates that all Oracle homes registered in the inventory are also shared. However, a shared inventory is contrary to Central Inventory design principles.

Consequently, **Oracle recommends placing the Central Inventory on a local file system not shared by other systems**, because the Central Inventory is a system-specific inventory of the installations on this system. Shared inventories are strongly discouraged.

Enforced Shared Inventory Check

Since a shared inventory is a recommended practice, OUI displays a warning if you attempt to specify a shared inventory location under the following circumstances:

- Grid Control upgrades from 10.1.x or 10.2.x to 10.2.0.4 or later

- 10.2.0.4 or later Grid Control patchset applications and 11g Grid Control installations

Exceptions are as follows:

- **Shared storage prior to 10.2.0.4** —

  If you have already set up your inventory on shared storage prior to Grid Control version 10.2.0.4, you can use the steps in "Procedure to Reconstruct the Central Inventory" below to reconstruct the central inventory on a non-shared location.

  If you do not want to reconstruct the central inventory on a non-shared location, you can pass the flag '-ignoreSharedInventory' to suppress the shared inventory error when invoking runInstaller.

- **HPTru64 Cases** —

  Tru64 is a special operating system where *all* storage is shared storage (including the operating system installed on shared storage). Therefore, a shared inventory is permissible.

Procedure to Reconstruct the Central Inventory

Perform the following steps as a Grid Control installation user, which is the user who owns the inventory.

1. Identify a non-shared location for the Central Inventory. This location should have sufficient permissions, such as read and write permissions. Make sure that the installation user creates this new location and that the location is part of the installation group.

2. Obtain the list of Oracle homes from the existing shared Central Inventory registered in oraInventory, use the command opatch lsinventory -all.

   Since this is a shared inventory, you may see Oracle homes listed in the Central Inventory from other systems on which the inventory is shared. Identify the Oracle home pertinent to the current system.

3. Edit the default inventory pointer file indicating the location of the non-shared inventory:

   ```
   inventory_loc=<location of non-shared inventory location>
   inst_group=<inventory group>
   ```

   The default inventory pointer file resides in this file:

   ```
   /etc/oraInst.loc
   ```

   All other Unix operating systems, such as Solaris and so forth reside in this file:

   ```
   /var/opt/oracle/oraInst.loc
   ```

4. For each Oracle home in the existing shared inventory, do the following:

   a. Detach the home from the existing inventory.

      – For a non-RAC home, run:

        ```
        $ORACLE_HOME/oui/bin/detachhome.sh
        ```

      – For a Clusterware, RAC, or clustered Agent home, run the following command on each of the nodes:

        ```
        $ORACLE_HOME/oui/bin/detachhome.sh -local
        ```

   b. Attach a home to the new non-shared inventory.

      – For a non-RAC home, run:

        ```
        $ORACLE_HOME/oui/bin/attachhome.sh
        ```

      – For a non-shared Clusterware home, run the following command on each of the nodes:

        ```
        $ORACLE_HOME/oui/bin/attachhome.sh -local "CLUSTER_NODES=
        {comma-separated nodelist}" LOCAL_NODE=<node on which command is run>
        CRS=true
        ```

      – For a shared Clusterware home, run the following command on each of the nodes:

        ```
        $ORACLE_HOME/oui/bin/attachhome.sh -local "CLUSTER_NODES=
        {comma-separated nodelist}" LOCAL_NODE=<node on which command is run>
        CRS=true -cfs
        ```

– For a non-shared RAC or clustered Agent home, run the following command on each of the nodes:

```
$ORACLE_HOME/oui/bin/attachhome.sh -local "CLUSTER_NODES=
{comma-separated nodelist}" LOCAL_NODE=<node on which command is run>
```

– For a shared RAC or clustered Agent home, run the following command on each of the nodes:

```
$ORACLE_HOME/oui/bin/attachhome.sh -local "CLUSTER_NODES={comma
separated nodelist}" LOCAL_NODE=<node on which command is run> -cfs
```

> **Note:** The old inventory becomes obsolete after this migration. You can delete the directory, but this is not mandatory.

Upgrading and Patching the Oracle Home

You can apply patchsets and upgrade an existing Oracle home. You can apply patchsets using Oracle Universal Installer. For more information on upgrading or applying patchsets for an Oracle product, refer to the respective Oracle product installation guide of the product that you want to upgrade.

Cloning Oracle Homes

You can clone an Oracle home using Oracle Universal Installer. For more information on cloning, see Chapter 6, "Cloning Oracle Software".

Backing up the Inventory

You can back up the Oracle home using your preferred method. You can use any method such as zip, tar, and cpio to compress the Oracle home.

It is highly recommended to back up the Oracle home before any upgrade or patch operation. You should also back up the Central Inventory when Oracle home is installed or deinstalled.

Example:

Consider a scenario where you have a Database Oracle home called DBHome that is registered with the default Central Inventory in the /product directory. You want to patch this database but decide to back up the database before patching.

1. Enter the following to back up the database:

```
cd /product/DBHome
tar cf - * | gzip > /product/archive/DBHome.tar.gz
```

If you are using a Win32 system, you could use WinZip to zip up the Oracle home. Do not use the jar command to zip the Oracle home, as this causes the file permissions to become lost.

2. Suppose you apply the patch and something goes wrong. You decide to delete the Oracle home from the Central Inventory and restore the original Oracle home. To delete the Oracle home from the Central Inventory, use the following command:

```
./runInstaller -silent -detachHome ORACLE_HOME="/product/DBHome"
```

On Win32 systems, the command would be:

```
setup.exe -silent -detachHome ORACLE_HOME="C:\product\DBHome"
```

3. Delete the Oracle home:

```
cd /product
rm -rf /product/DBHome
```

4. Restore the original Oracle home and update the Central Inventory. Restore the Oracle home to its original location using the following commands:

```
mkdir -p /product/DBHome
gunzip < /product/archive/DBHome.tar.gz | tar xf -
```

5. Attach this Oracle home to the Central Inventory:

```
./runInstaller -silent -attachHome ORACLE_HOME="/product/DBHome"
```

On Win32 systems, the command would be:

```
setup.exe -silent -attachHome ORACLE_HOME="C:\product\DBHome"
```

Recovering from Inventory Corruption

The inventory (Central and the Oracle home inventory) is critically important in the Oracle software life-cycle management. The following section explains what you need to do in case of inventory corruption.

Diagnosing and Recovering from Central Inventory Corruption

When you execute `opatch lsinventory -detail` or when you click **Installed Products,** the Oracle home does not appear.

Cause: The Oracle home may be missing from the Central Inventory, or the Central Inventory could be missing or corrupted.

Action: Do the following:

- If the Oracle home is missing from the Central Inventory, perform an attach home operation on the missing Oracle home. The Central Inventory will be restored.

- If the Central Inventory is missing or corrupted, restore the Central Inventory. If you have not backed up the Central Inventory, perform an attach home operation.

For more information on the attach home operation, see "Creating the Central Inventory" on page 2-8.

If multiple entries are in the `inventory.xml` file for a given Oracle home, Inventory Collection from the Grid Control perspective would have issues. In this event, you should remove these duplicate entries manually.

Here are some examples of the kind of incorrect entries that could be there:

```
<HOME NAME="db11g" LOC="/product/db11g" TYPE="O" IDX="1"/>
<HOME NAME="oms11g" LOC="/product/em/oms11g" TYPE="O" IDX="2"/>
<HOME NAME="db11g" LOC="/product/db11g" TYPE="O" IDX="3"/>
```

In the above example, the first and third entries are duplicates. The Oracle home name and Oracle home location are identical. In this example, remove the third line.

Note that all duplicate inventory issues are caused by manual updates to the `inventory.xml` file. Use the OUI APIs to change the inventory.

Diagnosing and Recovering from Oracle Home Inventory Corruption

When you execute `opatch lsinventory -detail` or when you click **Installed Products,** the Oracle home appears, but the products and components within the Oracle home are not listed.

Cause: This may result because of a missing or corrupted Oracle home inventory.

Action: If the Oracle home inventory is missing or corrupted, restore the Oracle home inventory. If you have not backed up the Oracle home inventory, you may have to install the software on a different node with the same platform and install the same patch levels including interim patches. After that, you can simply copy the inventory directory from the patched Oracle home to the location of the affected Oracle home.

> **Note:** For Oracle Universal Installer version 10.2.0.2.0 and above, you can use the following scripts in Oracle home to recover from Oracle home inventory corruption:
>
> - `detachHome.bat / detachHome.sh`: Use this script if the Oracle home is corrupted or needs to be updated.
>
> - `attachHome.bat / attachHome.sh`: Use this script if the Oracle home needs to be added to the inventory.

For information on recovering from Oracle Real Application Clusters Oracle home inventory corruption, see "Diagnosing and Recovering from RAC Oracle Home Inventory Corruption".

Oracle Real Application Clusters

In an Oracle Real Application Clusters environment, the inventory also contains a list of nodes associated with an Oracle home. It is important that during upgrade and patching, the inventory is correctly populated with the list of nodes.

For more information, see Chapter 5, "Installing Cluster Environments".

Updating the Nodes of a Cluster

When you use the `-updateNodeList` flag with the Oracle Universal Installer, it retrieves the list of nodes and updates the `inventory.xml` file. If the `Oracle Clusterware` tag is set to `TRUE`, the `Oraclehomeproperties.xml` file is updated with the Oracle Clusterware home information. For shared Oracle homes, you need to use this with the `-cfs` flag. The syntax is as follows:

```
./runInstaller -updateNodeList "CLUSTER_NODES={Node1,Node2}"
ORACLE_HOME="<Oracle_Home_Location>" -defaultHomeName LOCAL_NODE="Node_Name"
```

> **Note:** You update the nodes of a cluster only under exceptional circumstances. Oracle recommends that you perform this operation with the help of Oracle support.

Diagnosing and Recovering from RAC Oracle Home Inventory Corruption

When you execute `opatch lsinventory -detail` or when you click **Installed Products,** the Oracle home appears, but the products and components within the Oracle home are not listed.

Cause: This may result because of a missing or corrupted Oracle home inventory.

Action: Do the following:

1. Back up the Central and Local inventories of both Oracle Clusterware and RDBMS for all nodes.

2. Rename or remove the Central Inventory on all nodes.

3. Copy the Local Inventory from the node that is not corrupted to the node that is corrupted.

4. Recreate the Central Inventory with the attachhome option as in MetaLink note 413939.1, Steps to Recreate the Central Inventory in Oracle Real Application Clusters.

5. Verify that opatch lsinventory functions correctly in both the Oracle Clusterware and RDBMS homes.

> **Note:** Copying the Local Inventory is supported only when patch sets and one-off patches applied are the same across all nodes.

For information on recovering from non-RAC Oracle home inventory corruption, see "Diagnosing and Recovering from Oracle Home Inventory Corruption".

Home Selector (Available on Win32 Platforms)

The following sections describe the Home Selector, which is installed as part of Oracle Universal Installer on Windows computers.

To view the Home Selector, click the **Environment** tab of the Inventory dialog, which appears when you click the **Installed Products** button on several Oracle Universal Installer screens.

Home Selector Overview

The Home Selector is a part of the installation software. The Home Selector enables you to easily change your primary Oracle home (the one that appears first in the PATH environment variable). If you need to switch the active home or need to perform batch work which requires a "default home" to be active, you can use the Home Selector to change the Windows NT system settings.

When using the Home Selector to make a specific Oracle home the active one, the software installation in question is moved to the front of the PATH variable, making it the first directory to be scanned for executable and library files.

Use the GUI in the **Environment** tab of the Inventory dialog to establish the order of Oracle homes in your PATH variable.

How Home Selector Works

When you perform an installation on a system, Oracle Universal Installer runs the selectHome.bat file to register the Oracle home you selected. In silent mode, you perform this outside Oracle Universal Installer. The first Oracle home is named the "DEFAULT_HOME" and registers itself in the Windows NT registry under the key:

```
HKEY_LOCAL_MACHINE
    Software
        Oracle
```

This is the default Windows NT registry hive which contains all the "generic" Oracle settings. Also the PATH variable is adjusted and the BIN directory of the $ORACLE_HOME is added to the environment variable.

Some additional parameters is also written to the key:

```
HKEY_LOCAL_MACHINE
    Software
        Oracle
            KEY_<ORACLE_Home_Name>
```

With Oracle Universal Installer, you can change the "DEFAULT_HOME" name to something else. Also, it registers all Oracle settings in the "KEY_<Home_NAME>" sub-key of the ORACLE key.

When an additional Oracle home is added to the system, the PATH variable is adjusted again to add the new BIN directory of the newly installed Oracle software. The registry variables are written to a key named "KEY_<HOME_NAME>".

Oracle Home Directory Structure for Windows Platforms

A typical Oracle home on Windows platforms contains the files and directories shown in Table 2–5.

Table 2–5   Oracle Home Directory for Windows Platforms

Oracle Home Directory	Contents
\BIN	Product executable and DLLs
\LIB	DLL files
\JLIB	All JAR files
\DBS	Common message files
\PROD1	PROD1 product files
\PROD2	PROD2 product files

Optimal Flexible Architecture Directory Structure (on UNIX)

The Optimal Flexible Architecture (OFA) standard is a set of configuration guidelines for fast, reliable Oracle databases that require little maintenance.

OFA is designed to:

- Organize large amounts of complicated software and data on disk to avoid device bottlenecks and poor performance

- Facilitate routine administrative tasks such as software and data backup functions, which are often vulnerable to data corruption

- Alleviate switching among multiple Oracle databases

- Adequately manage and administer database growth

- Help eliminate fragmentation of free space in the data dictionary, isolate other fragmentation, and minimize resource contention

The OFA directory structure is described in the following sections.

ORACLE_BASE Directory

ORACLE_BASE specifies the BASE of the Oracle directory structure for OFA-compliant databases. A typical ORACLE_BASE directory structure is described in Table 2–6. When installing an OFA-compliant database using Oracle Universal Installer, ORACLE_BASE is set to /pm/app/oracle by default.

Table 2–6   Sample ORACLE_BASE Directory Structure and Content

Directory	Content
admin	Administrative files
doc	online documentation
local	Sub-tree for local Oracle software
product	Oracle software

ORACLE_HOME Directory

Table 2–7 shows an example of the Oracle home directory structure and content for an Oracle Server Installation.

If you install an OFA-compliant Oracle Server, the Oracle home directory is /mount_point/app/oracle/product/*release_number*. Under UNIX, the Oracle home directory might contain the following subdirectories, as well as a subdirectory for each Oracle product selected.

> **Note:**   The directory and content information in this table are only samples. The directories that appear in your own Oracle home depend on whether the corresponding products are installed and the version of the Oracle Database you are running.

Table 2–7   Sample Oracle Home Directory Structure and Content

Directory	Content
assistants	configuration Assistants
bin	binaries for all products
ctx	interMedia Text cartridge
dbs	init*sid*.ora, lk*sid*
install	install related files
lib	Oracle product libraries
jlib	Java classes
md	Spatial cartridge
mlx	Xerox Stemmer (for interMedia Text cartridge)
network	Net8
nlsrtl	NLS run-time loadable data
ocommon	common files for all products
odg	data gatherer
opsm	Parallel Server Manager Components
oracore	core libraries

Table 2–7   (Cont.) Sample Oracle Home Directory Structure and Content

Directory	Content
ord	data cartridges
otrace	Oracle TRACE
plsql	PL/SQL
precomp	precompilers
rdbms	server files and libraries required for the database
slax	SLAX parser
sqlplus	SQL*Plus

3

Customizing and Creating Response Files

This chapter introduces you to Oracle Universal Installer's availability of response files for silent and suppressed installations. This chapter also describes how to modify or create a response file so you can customize and standardize the installation of Oracle products in your organization.

This chapter is organized into the following sections:

- About Response Files
- Modifying a Response File
- Response File Format
- Installing with a Response File

About Response Files

This section answers the following questions:

- What Is a Silent Installation?
- What Is a Response File?
- Why Perform a Silent Installation?

What Is a Silent Installation?

A silent installation runs in the console and does not use the GUI. The interactive dialogs you normally see are not displayed in a silent installation.

Instead of prompting you to select a series of installation options, Oracle Universal Installer installs the software using a predefined set of options stored in a response file or passed on the command line. You can view the progress of a silent installation in the console.

What Is a Response File?

A **response file** is a specification file containing information you normally fetch through the Oracle Universal Installer user interface during an interactive installation session. Each answer is stored as a value for a variable identified in the response file.

For example, values for Oracle home or installation type can be set automatically within the response file.

The response file template for the installation of your product can be found on your stage (CD-ROM) under the `<root of CD>/response` directory, with pre-populated values. For example:

```
<Products.xml_Location>/Response/<product>.<installtype>.rsp
```

> **Note:** Check the installation guide for the product that you are installing to get the correct list of required files.

Why Perform a Silent Installation?

Silent installations can be useful if you have to install an Oracle product multiple times on multiple computers. If the options you select while installing on each computer are always the same, you save the time of reviewing each installation screen and selecting the various installation options.

Silent installations can also ensure that multiple users in your organization use the same installation options when they install your Oracle products. This makes supporting those users easier because you already know what components and options have been installed on each computer.

Before you perform a silent installation, you should review the settings in the response file template provided with your Oracle product.

> **Note:** If you attempt to perform a silent installation on a UNIX computer where no Oracle products have been installed, Oracle Universal Installer uses the default inventory location, then prompts you to run the oraInstRoot.sh script with root privileges upon successful installation.
>
> The script is saved in the _*Central Inventory - oraInventory *_ directory. This script sets up the Central Inventory on a clean host. You can override the default location by setting it in INVENTORY_ LOCATION. Note that this location is ignored if a Central Inventory already exists and is pointed to by /var/opt/oracle/oraInst.loc (or /etc/oraInst.loc for Linux, AIX, and Sequent).
>
> See "Oracle Universal Installer Inventory" on page 2-4 for more information on the Central Inventory.

The remainder of this chapter describes the various parameters and settings you can modify within an Oracle Universal Installer response file.

Modifying a Response File

If your product installation includes a response file template, you can find it on your stage (CD-ROM) under the `<root of CD>/response` directory.

If your product installation does not include a response file template, you can create a response file based on the installation options you select.

To modify the response file:

1. Make a copy of the product's response file and open it in a text editor.

2. Review any information provided in the response file or in the product installation guide.

Many software products use settings in the response file to customize the installation of their particular product. See the response file template for suggestions or guidelines on how to set up the file for your installation.

3. Get familiar with the organization and content of the response file using the information in the section "Response File Format" on page 3-3.

4. Modify the response file to meet the needs of your organization and save the modified version.

5. See the section "Installing with a Response File" on page 3-10 for information on starting Oracle Universal Installer using your modified response file.

Saving a Response File after Installation

You can use Oracle Universal Installer in interactive mode to record a response file, which you can edit and then use to complete silent mode or response file mode installations. This method is useful for custom or software-only installations. For more information, see "Recording a Response File" in the *Oracle® Database Installation Guide 11g for Linux*.

Response File Format

The following sections describe the organization and content of an Oracle Universal Installer response file.

- Variable Values

- Comments

- Headers

- Response File Parameters

> **Note:** Oracle recommends using an absolute path for the file name. However, if the file name is a relative path, it should be relative to the directory where the `oraparam.ini` file is present.

Variable Values

This section contains information on how variables are populated with values.

Variable Lookup Order

All variable values within a response file are in the name-value format. If two components have a variable with the same name, the expression should be written as follows to preserve each variable's uniqueness:

```
<component>:<variable>=<value>
```

Component Nomenclature and Version Ambiguity

If two components have the same internal name, but different versions, resolve the ambiguity by specifying the expression as follows:

```
<component>:<version>:<variable>=<value>
```

Order of Variables

Oracle Universal Installer looks for command line variables in the following order:

```
<component>:<version>:<variable>=<value>
<component>:<variable>=<value>
<variable>=<value>
```

Similarly, if command line variables are not found, Oracle Universal Installer looks for variables in the response file in the same order as shown above.

> **Note:** Oracle Universal Installer treats incorrect context, format, or type values within a response file as if no value were specified.

Response File Entries Order

There are no restrictions on where you place entries in the response file. You can insert entries in any order desired. You can also subsequently move existing entries to any other position within the file. Be aware, however, that although the order of the entries is unimportant, naming conventions require that the variable names must be unique regardless of where they appear in the file.

Setting the Recommendation Value

Values for variables are specified as:

```
<variable> = <recommendation> : <value>
```

The values that are given as `<value_required>` must be specified for a silent installation to be successful.

For values that are given as `<value_unspecified>`, you can optionally specify a value, where `<value>` can be one of the following types listed in Table 3–1.

Table 3–1   Variable Types and Representations

Type	Representation
Number	10
Boolean	TRUE or FALSE (case insensitive)
String	"Value"
String List	{"value1", "value2"}

The `<recommendation>` parameter can be specified as `Forced` or `Default`.

- If you specify `Forced`, no dialog appears during installation. The value is automatically used. You cannot change the value.

- If you specify `Default`, the dialog appears during installation with the value as the default. You can choose another value if desired.

- If nothing is specified for `<recommendation>`, `Forced` is used as the default.

Comments

Comments begin with a "#" (hash or pound) symbol. They contain information about the type of the variable, state whether the variable appears in dialog, and describe the function of the variable. A variable and a value are associated with a comment.

For example:

```
#TopLevelComponent;StringList;Used in Dialog
TopLevelComponent={"Demo", "1.0"}
```

Headers

The header is a comment that describes the different formats and comment conventions used in a response file.

For example:

```
#Parameter : UNIX_GROUP_NAME
#Type : String
#Description : UNIX group to be set for the inventory directory.
#Valid only on UNIX platforms.
#Example : UNIX_GROUP_NAME = "install"
UNIX_GROUP_NAME=<Value Unspecified>
```

Response File Parameters

The following list describes parameters for Oracle Universal Installer. All products installed using Oracle Universal Installer have these parameters in addition to product-specific variables.

> **Note:** The parameters provided below are OUI-specific and not necessarily the parameters applicable for your platform and version. Always check your specific installation guide for the list of valid response file parameters.

ACCEPT_LICENSE_AGREEMENT

Set this Boolean variable to TRUE if you agree with the license agreement. This parameter requires you to set this variable to TRUE to indicate your acceptance of the license agreement normally accepted in the GUI. This variable is used only in silent installations. Installation cannot continue until this variable is set to TRUE.

CLUSTER_NODES

This parameter lists the nodes on the cluster on which the Oracle home is installed or would be installed. For new installations, this would be the node list on which the installation needs to be done (including the local node). For patching or upgrades, this is the node list on which the Oracle home is already installed.

```
"CLUSTER_NODES={alpha-1, alpha-2}"
```

DECLINE_SECURITY_UPDATES

If the direct connection is unavailable, set this parameter to false if you do not want to configure Oracle Configuration Manager, or do not want to provide any proxy details for the connection. The default is False.

If you do not set MYORACLESUPPORT_USERNAME to any value and you set this parameter to true, the Oracle Configuration Manager is not configured.

For information on the Oracle Configuration Manager, see the *Oracle® Configuration Manager Installation and Administration Guide*.

DEINSTALL_LIST

Use this parameter to enter a list of components to be removed during a silent deinstall session. For each component, specify the internal component name and version.

```
DEINSTALL_LIST={"sample1","1.0.0.0.0"}
```

DEPENDENCY_LIST

The DEPENDENCY_LIST is the list of dependents on which the component depends. These dependent components represent the list of components that appear as "selected" during installation. Following is a list of some of this parameter's characteristics:

FROM_LOCATION

The location of the source of the products to be installed. The default generated value for this parameter is a path relative to the location of the runInstaller file oraparam.ini. Relative paths are necessary for shared response files used by multiple users over a network. Since users may be mapping to different drives, absolute paths will not work for shared response files.

> **Notes:**
>
> ■ This parameter serves the same purpose as SOURCE in oraparam.ini. Use this parameter only if you need to override the default products.xml location.
>
> ■ You must enter a value for FROM_LOCATION for a complete silent installation. You may want to use the command line to set this parameter. If the location is a relative path, remember that the path should be relative to the location of the oraparam.ini file.
>
> ■ Make sure that /var/opt/oracle/oraInst.loc (or /etc/oraInst.loc for Linux, AIX, and Sequent) exists. The response file for UNIX has a public variable, setunixinstallgroup, that only becomes effective when it is the first Oracle Universal Installer installation and the oraInst.loc file is not already present in the location mentioned above.

FROM_LOCATION_CD_LABEL

This parameter is used in multiple-CD installations. It includes the label of the compact disk where the file products.jar exists. You can find the label in the disk.label file in the same directory as products.jar.

INCLUDE

This parameter specifies the list of response files to be included in this response file. If you want to include other response files in your main response file, you must use this parameter. Note that the values in the main response file have priority over the included response files.

```
INCLUDE={"file1.rsp","file2.rsp",............."filen.rsp"}
```

You should specify the absolute path in the INCLUDE statement for each response file to be included. If you wish to specify a relative path, note that the location is relative to the current working directory (the location of the oraparam.ini file).

INSTALL_TYPE

You can set the installation type variable to determine the installation type of the currently selected top level component.

The installation type variable is only generated for the top level components and only when there is more than one installation type available.

> **Note:** You must enter a value for INSTALL_TYPE.

LOCAL_NODE

This parameter specifies the current node on which the installation occurs.

`LOCAL_NODE={}`

LOCATION_FOR_DISK2

This parameter is for the complete path to other disks.

`LOCATION_FOR_DISK2="F:\teststage\cd\Disk2"`

If there are more than two disks, more variables will be added as `LOCATION_FOR_DISK3`, and so on.

The CD location for a silent installation is located by two mechanisms:

1. Oracle Universal Installer looks for the `Location_For_Disk{DiskNumber}` variable in the response file and uses that location.

2. If the variable does not have a value or does not have the required files, it looks for the components under `../../Disk{DiskNumber}/stage` (from `products.jar`).

    Therefore, you can either specify the location to look for it in the response file or you can copy them into the disk. For example:

    `<Product_Shiphome_Location>/stage`

NEXT_SESSION

Set this boolean variable to `TRUE` if the installer needs to go to the **File Locations** page for another installation or to another response file if you are performing a silent installation.

`NEXT_SESSION = TRUE`

NEXT_SESSION_ON_FAIL

Set this boolean variable to `TRUE` to enable users to invoke another installation session even if the current installation session fails. This variable is used only if the `NEXT_SESSION` variable is set to `TRUE`.

`NEXT_SESSION_ON_FAIL = TRUE`

NEXT_SESSION_RESPONSE

Use this string variable to enter the complete path of the next session's response file if you want to automatically begin another silent installation when the current installation is complete. If you specify only a file name, the Installer looks for the response file in the `<TEMP>/orainstall` directory. This variable is only used if `NEXT_SESSION` is set to `TRUE`; otherwise, Oracle Universal Installer ignores the value of this variable.

`NEXT_SESSION_RESPONSE="/private/usr2/nextinstall.rsp"`

MYORACLESUPPORT_PASSWORD

This parameter indicates the My Oracle Support password used in Oracle Configuration Manager (OCM) configuration. This parameter is used for validation only if the direct connection is available, or if there is a valid proxy connection (in case no direct connection is available). The default is Empty.

For information on OCM, see the *Oracle® Configuration Manager Installation and Administration Guide*.

MYORACLESUPPORT_USERNAME
This parameter indicates the My Oracle Support username/email address used in Oracle Configuration Manager (OCM) configuration. This parameter validates the My Oracle Support user name only if the direct connection is available, or if there is a valid proxy connection (in case no direct connection is available).

To avoid configuring OCM, you do not need to pass any values for MYORACLESUPPORT_USERNAME or any other session variables except for DECLINE_SECURITY_UPDATES, which should be set to true. OCM will then be installed, but not configured.

Configuration Use Cases

- If only MYORACLESUPPORT_USERNAME is provided,

    - OUI does not validate the user name against My Oracle Support, but configures it in anonymous email registration mode.

    - there is not a direct connection, and DECLINE_SECURITY_UPDATES=true, OUI does not validate the user name against My Oracle Support, but configures it in disconnected mode.

    - there is not a direct connection, and PROXY_* values are provided, OUI does not validate the user name against My Oracle Support, but configures it in anonymous email registration mode.

    - SECURITY_UPDATES_VIA_MYORACLESUPPORT=true, and MYORACLESUPPORT_PASSWORD is provided, OUI validates it against My Oracle Support and configures it in My Oracle Support user registration mode.

    - SECURITY_UPDATES_VIA_MYORACLESUPPORT=true, and MYORACLESUPPORT_PASSWORD is provided, but there is not a direct connection and DECLINE_SECURITY_UPDATES=true, OUI configures in disconnected mode.

    - SECURITY_UPDATES_VIA_MYORACLESUPPORT=true, MYORACLESUPPORT_PASSWORD is provided, but there is not a direct connection and PROXY_* values are provided, OUI configures in My Oracle Support user registration mode.

- If MYORACLESUPPORT_USERNAME is empty and DECLINE_SECURITY_ UPDATES = true, OUI does not configure OCM.

For information on OCM, see the *Oracle® Configuration Manager Installation and Administration Guide*.

OPTIONAL_CONFIG_TOOLS
While all the required configuration tools are launched by the installer, you can control the optional configuration tools you would want to launch by specifying the tool's internal names in the `OPTIONAL_CONFIG_TOOLS` section.

> **Note:** The `OPTIONAL_CONFIG_TOOLS` variable is only generated when at least one optional configuration tool is available.

You can call the Net Configuration Assistant or the Database Configuration Assistant at the end of a database installation in silent mode also. For example, to launch the Net Configuration Assistant in silent mode, you can pass the parameter `s_ responseFileName="netca.rsp"`.

You can specify both the Auto-launch optional tools and User-launch optional tools in a string list.

```
#Example : OPTIONAL_CONFIG_TOOLS = {"configtool2 ", "configtool3"}
OPTIONAL_CONFIG_TOOLS=<Value Unspecified>
```

If no value is specified for this variable, all the tools are run by default. If there is a value specified, only those optional tools are run while the rest of the tools are ignored.

Suppress the configuration tool by mentioning only the tools that you want to run as part of the OPTIONAL_CONFIG_TOOLS variable added for each component. You should use the internal names of the configuration tool. The response file generator generates these internal names also as part of the options provided for the variable.

For example, if oracle.server has Tool1 and Tool2 and you want to run only Tool1 in the response file, you can specify it as follows:

```
oracle.server:11.2.0.0.0:OPTIONAL_CONFIG_TOOLS={"Tool1"}
```

ORACLE_BASE
The location of the Oracle Base where all Oracle software and configuration-related files are stored.

ORACLE_HOME
The location where products are to be installed. You must enter a value for ORACLE_HOME for a complete silent installation.

PROXY_HOST
This proxy connection server is used only if the direct connection is not available and DECLINE_SECURITY_UPDATES=false.

PROXY_PORT
This proxy connection port is used only if the direct connection is not available and DECLINE_SECURITY_UPDATES=false.

PROXY_PWD
This proxy connection password is used only if the direct connection is not available and DECLINE_SECURITY_UPDATES=false.

PROXY_USER
This proxy connection user name is used only if the direct connection is not available and DECLINE_SECURITY_UPDATES=false.

REMOTE_NODES
This parameter lists the remote nodes (apart from the local node) on the cluster on which to install during a silent installation. Oracle Universal Installer installs on all named nodes.

```
"REMOTE_NODES={alpha-1, alpha-2}"
```

REMOVE_HOMES
Use this parameter to identify the Oracle homes you want to remove from the inventory during a deinstallation session. For each home, specify the home name using the full path information.

```
REMOVE_HOMES={"/home/oracle/ora9i","/home/oracle/ora8i"}
```

RESPONSEFILE_VERSION
This parameter specifies the version number of the response file.

```
RESPONSEFILE_VERSION = <version_number>
```

RESTART_SYSTEM

Set this boolean variable to `TRUE` if you want to restart the system without the user's confirmation. This is the force value for restarting the system.

SECURITY_UPDATES_VIA_MYORACLESUPPORT

This parameter is equivalent to the OUI window checkbox that enables users to receive security updates. ("I wish to receive security updates via My Oracle Support," followed by a My Oracle Support Password field.) This parameter enables you to set the password for My Oracle Support credentials. The default is False. If you set this to true, the user must enter a password.

SELECTED_LANGUAGES

You can set the languages in which the components will be installed. You must use the internal name while specifying the value:

```
en,    : English
fr,    : French
de,    : German
ja,    : Japanese
```

For example, to specify Japanese:

```
SELECTED_LANGUAGES = {"ja"}
```

For more information on the languages Oracle Universal Installer supports, see "Product Language Selections" on page 8-1.

> **Note:** The `SELECTED_LANGUAGES` variable is only generated when more than one language is available.

TOPLEVEL_COMPONENT

This parameter is the name of the component (products) and the version as a string list. You must enter a value for `TOPLEVEL_COMPONENT`.

Usually, the components are represented with a pair of strings: the first one representing the internal name, the second representing the version.

For example, RDBMS 11.2. may be represented as `{"oracle.rdbms","11.2.0.0.0"}`.

UNIX_GROUP_NAME

This parameter is the UNIX group name to be set for the inventory on UNIX platforms.

Note: The UNIX group name is used for first-time installations only.

Installing with a Response File

Many Oracle software products provide tools and procedures for running Oracle Universal Installer from the command line without displaying Oracle Universal Installer screens or responding to questions during the installation.

These are called silent installations. Instead of prompting you to select a series of installation options, Oracle Universal Installer installs the software using a predefined set of options. These options are stored in a response file (`.rsp`).

Consider the following information about response files:

- If your product installation includes a response file, you can find it on your stage (CD-ROM) under the `<root of CD>/response` directory.

- You can modify the response file for your Oracle product to customize an installation for your organization. See "Modifying a Response File" on page 3-2 for more information.

- If you start Oracle Universal Installer from the command line, see "Using Oracle Universal Installer Exit Codes" on page 4-11.

The following sections describe how to specify a response file when you start Oracle Universal Installer.

> **Note:** If you attempt to perform a silent installation on a UNIX computer where no Oracle products have been installed, you will receive an error message. Before you can perform a silent installation on such a computer, you must first run the script `oraInstRoot.sh`, which is saved in the `/oraInventory` directory. You must run this script with root privileges. This enables Oracle Universal Installer to set up the Central Inventory on a clean host.
>
> See "Oracle Universal Installer Inventory" on page 2-4 for more information on the Central Inventory.

Specifying a Response File

To start Oracle Universal Installer and specify the response file, enter the following command at the command line in the directory where the executable file is installed:

```
setup.exe -responseFile <filename> <optional_parameters> (on Windows)
./runInstaller -responseFile <filename> <optional_parameters> (on UNIX)
```

> **Note:** You must specify the complete `responseFile` path. If you do not, Oracle Universal Installer assumes the location to be relative to the `oraparam.ini` associated with the launched Oracle Universal Installer.

For help on command line usage, enter the following at the command line in the directory where the executable file is stored:

```
setup -help (on Windows)
./runInstaller -help (on UNIX)
```

In Windows, when you execute `setup -help`, a new command window appears, with the "Preparing to launch..." message. A moment later, the help information appears in that window.

Optional Parameters When Specifying a Response File

Optional parameters you can use with the `-responseFile` flag are:

- **-nowelcome** — Use the `-nowelcome` flag with the `-responseFile` flag to suppress the Welcome dialog that appears during installation.

- **-silent** — Use the `-silent` flag with the `-responseFile` flag to run Oracle Universal Installer in complete silent mode. Note that the Welcome dialog is suppressed automatically.

- **-formCluster** — Use the `-formCluster` flag for Oracle Clusterware installations to specify the cluster.

Note that when you specify the `-silent` flag, but no values are specified to a particular variable in a dialog, the installer stops. The success or failure of the installation when you specify this flag is generated as follows:

- In a file named `silentInstall<timestamp>.log` for hosts without an Oracle inventory. This file is generated in the `/tmp` directory on UNIX and the directory specified by the variable `TEMP` on Windows platforms.

- In the inventory logs directory for hosts that already had an inventory.

> **Note:** Using the `-nowelcome` option with the `-silent` option is unnecessary since the Welcome screen does not appear when you use the `-silent` option.

Setting Response File Variables From the Command Line

With Oracle Universal Installer 2.1 and higher, you can specify the value of certain variables when you start Oracle Universal Installer from the command line. Specifically, you can specify session and component variables. For specific information about the format and organization of response files, see "Modifying a Response File" on page 3-2.

When you specify the value of a variable on the command line, that value overrides the value of the variable if it is defined in the response file.

Specifying the Value of a Session Variable

To specify the value of a session variable, use the following command syntax:

```
./runInstaller session:<variable_name>=<value> (on UNIX)
setup.exe session:<variable_name>=<value> (on Windows)
```

For example, to install English and French files:

```
./runInstaller session:COMPONENT_LANGUAGES="{"en","fr"}" (on UNIX)
setup.exe session:COMPONENT_LANGUAGES="{"en","fr"}" (on Windows)
```

Note that the "session:" tag is optional and is used mainly to remove any possible ambiguity.

Specifying the Value of a Component Variable

To specify the value of a component variable, use the following command syntax:

```
./runInstaller <component_name>:<component_version>:<variable_name>=<value> (on UNIX)
setup.exe <component_name>:<component_version>:<variable_name>=<value> (on Windows)
```

For example, to modify the value of a variable `VAR1` in version 1.1 of a component called `COMP2`:

```
./runInstaller COMP2:1.1.0.0.0:VAR1="test" (on UNIX)
setup.exe COMP2:1.1.0.0.0:VAR1="test" (on Windows)
```

Note that the "`<component_name>:`" and "`<component_version>:`" optional tags are used mainly to remove any possible ambiguity. If two variables exist with the same name for different components/versions, use these tags to distinguish between them.

4

Installing Products

This chapter includes the following sections:

- Checking Prerequisites Before Installation
- Installing Oracle Products
- Deinstalling Oracle Products
- Running Oracle Universal Installer After Installation
- About Oracle Universal Installer Log Files

Checking Prerequisites Before Installation

Before installation, Oracle Universal Installer checks the environment to see whether it meets the requirements for successful installation. Early detection of problems with the system setup reduces the chances of encountering problems during installation; for instance, problems with insufficient disk space, missing patches, inappropriate hardware, and so on.

Oracle Universal Installer is required to perform all prerequisite checks defined for the installation before installing any software, whether they are Oracle Universal Installer-specific tests, or tests defined for a specific product. Specific prerequisite checks are defined for each operating system on which Oracle Universal Installer runs. All prerequisite check parameters must be defined in the `oraparam.ini` file (or another `*.ini` file that you define). All the results are logged in the `install Actions<timestamp>.log` file.

You can perform prerequisite checking automatically when you run the Oracle Universal Installer executable during an installation. Simply run Oracle Universal Installer to perform all predefined prerequisite checks.

Inputs to the checker are listed in the `prerequisite.xml` file. After running the checker, you can find the results, along with the predefined inputs, in the `prerequisite_results.xml` file. These files are located in the `oraInventory/logs` directory. You can reuse the `prerequisite_results.xml` file as an input file for subsequent executions of the checker.

Installing Oracle Products

The following sections describe how to start Oracle Universal Installer and install an Oracle product. Specifically, this section describes:

- Getting Help While Installing Oracle Products
- About the ORAPARAM.INI File

- [Modes of Installation](#)
- [Installation Media](#)
- [Special Instructions for UNIX Users](#)

Getting Help While Installing Oracle Products

At any time while installing your product, click **Help** for information about the screens specific to your installation.

Oracle Universal Installer provides two kinds of online help

- Generic online help provided with every copy of Oracle Universal Installer

  These topics describe the screens and dialog boxes that all Oracle Universal Installer users see, regardless of the product they are installing.

- Online help specific to a particular installation

  These topics are created by the product developer and describe the screens and dialog boxes specific to the product you are installing. For example, the help topic for the **Installation Types** page is often a custom help topic the installation developer creates that describes the specific installation types for the product you are installing.

After you view an online help topic, choose **Navigator** from the **Tools** menu to display the navigator pane. From the navigator pane, you can browse the table of contents, select other topics, or search for a particular word or phrase in the online help.

> **Note:** Only generic help topics are available in the navigator or table of contents. You can only access custom help topics by clicking the Help button on the dialogs or windows they describe.

About the ORAPARAM.INI File

The `oraparam.ini` file is the initialization file for Oracle Universal Installer. This file includes information that defines the behavior of certain Oracle Universal Installer features. Each product installation possesses a unique `oraparam.ini` file.

Generally, you should not have to edit the contents of this file, but in certain situations, understanding the contents of this file can help you troubleshoot problems and understand certain aspects of the Oracle Universal Installer product.

For example, for most installations, Oracle Universal Installer provides a default value on the **File Locations** page that points to the location of the product's installation kit or stage. This default value is stored in the `oraparam.ini` file. The `oraparam.ini` file also identifies the location of the Java Runtime Environment (JRE) required for the installation.

In the staging area, it is located in the same directory as the executable file. For example:

For UNIX systems:

```
cd/<Product_Shiphome_Location>/install/solaris/runInstaller
cd/<Product_Shiphome_Location>/install/solaris/OraParam.ini
```

For Windows systems:

```
cd\<Product_Shiphome_Location>\install\win32\setup.exe
cd\<Product_Shiphome_Location>\install\win32\OraParam.ini
```

In the staging area, the default OUI_LOCATION is relative to the location of the oraparam.ini file, as follows:

```
../../stage/
```

Once installed, the oraparam.ini file is located in the /oui directory.

Table 4–1 describes the parameters in the oraparam.ini file and how to use them.

Table 4–1   Parameters in oraparam.ini

Section/Parameter	Description
[Oracle]	This section contains various parameters related to your installation.
DISTRIBUTION	Set to TRUE if the oraparam.ini file is on a distribution medium. A distribution medium can be a CD-ROM or a Web server. Once installed, this parameter is set to FALSE.
SOURCE	Location of your staging area (the products.xml file). This location is relative to the directory where oraparam.ini exists.
LICENSE_LOCATION	Location of a text file for License information. This location is relative to the directory where oraparam.ini exists. The legal terms file should be a plain text file.
	If you specify the LICENSE_LOCATION variable, Oracle Universal Installer asks for license acceptance after you click the Next button on the "Welcome" screen. Oracle Universal Installer lets you proceed only after the license is accepted.
	This parameter is ignored if the file is not found.
LICENSE_TITLE	The value of this variable displays as the title of the license agreement. Oracle Universal Installer only reads this value if the license dialog is displayed; that is, if the LICENSE_LOCATION variable has a valid value.
JRE_LOCATION	Location of the Java Runtime Environment (JRE) that the Oracle Universal Installer uses.
	Note: For a single installation to point to two different JRE versions, move the oraparam.ini file one level below its original location and then specify the individual platform's JRE location. For example, if the original location is in the Disk1/install directory, move it to the Disk1/install/win32 or the Disk1/install/solaris directory, and then specify the individual platform's JRE location.
OUI_LOCATION	Points to the location of Oracle Universal Installer files used for interactive mode (GUI-based). This parameter is optional; if DISTRIBUTION=TRUE, Oracle Universal Installer computes this value using the OUI_VERSION parameter.
	Use this parameter if you want to override the default value:
	`../../stage/Components/oracle.swd.oui/<version>/1/DataFiles/Expanded`
OUI_CORE_LOCATION	Points to the location used for silent mode. This parameter is optional; If DISTRIBUTION=TRUE, then Oracle Universal Installer computes this value using the OUI_VERSION parameter.
	Use this parameter if you want to override the default value:
	`../../stage/Components/oracle.swd.oui.core/<version>/1/DataFiles/Expanded`

Table 4–1 (Cont.) Parameters in oraparam.ini

Section/Parameter	Description
OUI_VERSION	Set the version of Oracle Universal Installer that you are using. You must properly set the version for the BOOTSTRAP to work.
DISPLAY_VERSION	Set to FALSE to suppress the display of the version of top-level components in the Installation Type dialog during installation.
JRE_MEMORY_ OPTIONS	Set these to increase the initial heap threshold for JRE. For example, -mx48m.
DEFAULT_HOME_ LOCATION	Location of the default Oracle home.
DEFAULT_HOME_ NAME	The default name for the Oracle home. Use this parameter only if the installation occurs on a host with no previous Oracle installations.
NO_BROWSE	Lists directories that you do not want to browse, which are typically large directories that require a long time to view. For example, /net, /nfs.
NLS_ENABLED	Set to TRUE for Oracle Universal Installer to enable NLS support. Set to FALSE to disable the installation session translations. Oracle Universal Installer displays in English even if you run on a non-English system.
BOOTSTRAP	Set to TRUE or FALSE. This parameter instructs Oracle Universal Installer to attempt a bootstrap. Set to TRUE before cutting CDs, but set to FALSE after you have copied the staging area to the hard disk.
BOOTSTRAP_SIZE	Use this parameter to set the size the temporary space requires when BOOTSTRAP is set to TRUE. For example, when you install Oracle Universal Installer, it sets the value of this parameter to the temporary space required by both Oracle Universal Installer and the JRE.
	By default, if this entry is not set, Oracle Universal Installer assumes 45 MB for Win32, 52 MB for Win64, and 69 MB for Solaris. However, these values could vary from one major release to the other, based on the space required by newer versions of JRE.
	If the shiphome contains advertisement images, installation developers should add the space taken by the images to this value. Oracle Universal Installer checks the temp space requirements before starting up and produces an error if there is not enough space for Oracle Universal Installer to run in bootstrap mode.
USE_BUILD_ NUMBER	Use this parameter to control whether or not Oracle Universal Installer considers the build number of the component when determining whether or not to overwrite a previous version or copy of a component. Note that this parameter is intended to be used in pre-production shiphomes only.
APPLTOP_STAGE	Use this parameter when you want your users to specify the location of an Oracle Applications top (APPL_TOP) directory. When this parameter is set to TRUE, the **File Locations** page includes fields for selecting an APPL_TOP directory. This is an optional parameter for use with Apps installs only. If not specified, the default is assumed to be FALSE.
REGISTRATION_URL	This parameter is the URL where the user is directed from the Product Registration page in Oracle Universal Installer. The REGISTRATION_KEY parameter is validated at this site. Set this parameter along with the REGISTRATION_KEY parameter to invoke the Product Registration page. Both are required.

Table 4–1   (Cont.) Parameters in oraparam.ini

Section/Parameter	Description
REGISTRATION_KEY	This key is validated against an encrypted key at the `REGISTRATION_URL` location. Set this parameter along with the `REGISTRATION_URL` parameter to invoke the Product Registration page. Both are required.
[IMAGES]	This section lists the images associated with an installation. To show advertisements during an installation, specify each image as a separate variable.

Modes of Installation

You can use Oracle Universal Installer to install Oracle products in any of the three following modes:

- **Interactive:** Use Oracle Universal Installer's interactive mode to use the graphical user interface to walk through the installation, providing information in the installation dialogs when prompted. This method is most useful when installing a small number of products in different setups on a small number of hosts.

- **Suppressed:** Use Oracle Universal Installer's suppressed mode to supply the necessary information by using a combination of a response file or command line entries with certain interactive dialogs. You can choose which dialogs to suppress by supplying the information at the command line when you invoke Oracle Universal Installer. This method is most useful when an installation has a common set of parameters that can be captured in a response file, in addition to custom information that must be input by hand.

- **Silent:** Use Oracle Universal Installer's silent installation mode to bypass the graphical user interface and supply the necessary information in a response file. This method is most useful when installing the same product multiple times on multiple hosts. By using a response file, you can automate the installation of a product for which you know the installation parameters. For more information, see Chapter 3, "Customizing and Creating Response Files", for detailed information on using response files and installing in silent mode.

> **Note:**   You can use the `-noConsole` flag on the Windows platform to suppress the display of messages in the console.

Installation Media

For each of these three installation modes, you can install from three different media:

- Installing from a Single CD-ROM

- Installing from Multiple CD-ROMs

- Installing from a staged HTTP location

> **Note:**   On Windows, when you start the installer from a shared drive, you need to map the shared drive and then invoke the installer from the shared drive.

The following sections discuss these different installation approaches.

> **Notes:** When you invoke `runInstaller` (UNIX) or `setup.exe` (Windows), you should invoke it from the directory where this command is present, or you must specify the complete path to `runInstaller` (UNIX) or `setup.exe` (Windows).

Installing from a Single CD-ROM

While installing Oracle products contained on a single CD-ROM, start Oracle Universal Installer by running the executable file, `setup.exe` or `runInstaller.sh,` located in:

`../install/<platform>`

Where `<platform>` represents Win32, Win64, Solaris, Linux, and so on.

For UNIX systems, run the script by typing `./runInstaller` at the command line.

> **Note:** Oracle Universal Installer for win64 functions like Oracle Universal Installer for win32. However, the startup directory on the CD is "win64" instead of "win32." Launching Oracle Universal Installer from the win32 directory launches Oracle Universal Installer in 32-bit mode, used for installing 32-bit software. Use win64 for installing 64-bit software.
>
> When you install both 32-bit and 64-bit Oracle Universal Installer on a 64-bit machine, two different inventories are created and maintained. However, you cannot install 64-bit software in a 32-bit home, and vice versa.

Installing from Multiple CD-ROMs

If you are creating a multiple-CD installation on UNIX, you might need to launch `runInstaller` in the background using the following command:

`./runInstaller &`

By launching `runInstaller` in the background, you can change your current directory after you launch Oracle Universal Installer, enabling you to eject the CD. (It may also help to launch `runInstaller` as a foreground process from a different directory.)

You may want to create a shell script that launches Oracle Universal Installer in the background and then exits. If you choose to create a shell script, remember to also pass all parameters that you passed to the shell script to `runInstaller` in the event that you want to install silently using a response file.

TEMP/TMP Directory On both UNIX and Windows installations, temporary copies of Oracle Universal Installer and JRE are placed in the `TEMP` or `TMP` directory in a subdirectory named `/OraInstall<timestamp>` so that these applications can be launched when you change CD-ROMs. Note that temporary files are created for single-CD installations as well. On both UNIX and Windows, Oracle Universal Installer looks for `%TEMP%` then `%TMP%`. If neither is set, Oracle Universal Installer defaults to `/tmp` on UNIX and `c:\temp` on Windows

> **Note:** .The `TEMP/TMP` directory should not be a cluster file system or a shared location.

Unmounting a CD On UNIX, if you have trouble installing a product from multiple CD-ROMs, try using the following procedure to unmount the first CD-ROM and mount the second CD-ROM. If you still have problems, refer to the documentation links at the end of this topic.

In most cases, the following procedure helps with any problems you experience while switching to a second CD-ROM while installing Oracle software. If you inadvertently run the installer while the current working directory is in the CD-ROM, follow these steps to mount the next CD-ROM:

1. Change to the root directory of your system and log in as the root user by using the following commands:

   ```
   $ cd /
   $ su root
   ```

2. Unmount and remove the CD-ROM from the drive with the following command:

   ```
   # umount cdrom_mount_point_directory
   ```

3. Insert and mount the next CD-ROM into the drive by using the following command:

   ```
   # mount options device_name cdrom_mount_point_directory
   ```

4. Enter the correct mount point in the Installation dialog box.

5. Click **OK** to continue.

If after attempting this procedure you are still having problems, see the section on installing from multiple CD-ROMs in the *Oracle Database Installation Guide*, which is available from the Oracle Technology Network:

```
http://otn.oracle.com/documentation
```

Installing from a staged HTTP location

With Oracle Universal Installer, you can install products from the Web. You can publish your staging area from a Web server and then in the Oracle Universal Installer's **Source** location, specify the HTTP location for the `products.xml` file.

For example, you can enter:

```
http://www.oracle.com/product/ouiinstall/stage/products.xml
```

The Oracle Universal Installer recognizes a Web staging area just like a local, network, or CD-ROM stage.

System administrators of large customers who may want to deploy Oracle software to more than one target can use a combination of the Web installation and response file features:

1. Copy the staging area to a shared file system and make it accessible on the Intranet or a Web server.

2. Include predetermined response files on the same location. (Different groups of users might rely on different response files.)

3. Clients run Oracle Universal Installer locally and use the local response file that is mailed or downloaded so they can perform a silent installation.

The Web installation capability relies on some guidelines that must be followed at installation development time. Check the installation guide for your product to see if the installation of your product is certified for Web installation.

To test if your stage is Web-enabled, you can try the following procedure:

1. Copy the stage to your Web server.

2. Start the Oracle Universal Installer locally and point to the location of the `products.xml` file. For example:

```
http://smpweb.us.oracle.com/product/ouiinstall/stage/products.xml
```

Special Instructions for UNIX Users

The following sections describe special instructions that apply when you are installing certain products on a UNIX system.

Failed to Connect to Server Error

If you receive an Xlib error or a "Failed to connect to Server" error when you are running Oracle Universal Installer on the Solaris operating system, do the following:

1. Define the following environment variable on the host computer where you are running Oracle Universal Installer:

```
%setenv DISPLAY <machine name>:0.0
```

2. Replace `<machine name>` with the name of the computer that will display Oracle Universal Installer.

3. On the computer that will display Oracle Universal Installer, enter the following command, which enables other computers to display information on the computer's monitor:

```
%xhost +
```

4. Rerun the `runInstaller` script after you have set the `DISPLAY` environment variable.

> **Note:** You can run Oracle Universal Installer without specifying the `DISPLAY` variable by running in silent mode using a response file.

Providing a UNIX Installer Location with Root Privileges

You must have root privileges to perform various installation operations on the UNIX platform. For example, you must have root privileges to be able to create the Oracle Universal Installer inventory.

If you are installing Oracle Universal Installer for the first time, you are prompted to run a shell script from another terminal window before proceeding with the installation. Oracle Universal Installer prompts you to run `root.sh` after installation completes *only* if the script is required to be run as `root` before configuration assistants are run. Otherwise, you are prompted to run `root.sh` as `root` later.

> **Note:** When running Oracle Universal Installer in silent mode, if `root.sh` is required prior to configuration assistants, Oracle Universal Installer skips configuration assistants during the installation. You must run `root.sh` as `root` and then run the skipped configuration assistants after the silent installation is complete.

To successfully run the required shell script:

1. Leave the Oracle Universal Installer window open and open another terminal window.

2. In the new terminal window, use the substitute user command to log in with root privileges:

    ```
    su -root
    ```

3. Change directory to the Oracle home into which you are currently installing your Oracle software product.

4. Run the shell script `./root.sh`.

5. When the script is finished and you are returned to the command prompt, exit from the new terminal window and return to Oracle Universal Installer to continue the installation.

> **Note:** Do not exit the installation to run the shell script. Exiting the installation removes this script.
>
> You are prompted to run the script only the first time you install.

Providing a UNIX Group Name

If you are installing a product on a UNIX system, the Installer also prompts you to provide the name of the group that owns the base directory.

You must choose a UNIX group name that has permissions to update, install, and remove Oracle software. Members of this group must have write permissions to the base directory chosen.

Only users who belong to this group are able to install or remove software on this host.

Deinstalling Oracle Products

You can only deinstall Oracle products by using the Deinstallation tool. For more information, see see "Removing Oracle Database Software" in the *Oracle® Database Installation Guide 11g for Linux*.

Running Oracle Universal Installer After Installation

The following sections describe the different ways that Oracle Universal Installer can be used after installation. Specifically, this section describes:

■ Starting Oracle Universal Installer

■ Command Line Arguments

■ Using Oracle Universal Installer Exit Codes

■ Cloning Considerations

Starting Oracle Universal Installer

OUI is installed in the Oracle home and is available for both Windows and Unix under:

```
<Oracle home>/oui/bin
```

For all platforms, the executable file (`setup.exe` or `runInstaller.sh`) is located in the following directory:

```
<oui_location>/bin
```

A new version of Oracle Universal Installer replaces its older version.

To start Oracle Universal Installer:

- On Windows platforms, select **Start**, **Programs**, **Oracle Installation Products**, **Oracle Universal Installer.**

- On UNIX, execute `./runInstaller` from the directory where it is installed.

  For example: if the Oracle home is /u01/app/oracle/, OUI will be located at OH/oui/u01/app/oracle/oui.

A `runInstaller.sh` script is also available, so that you can launch Oracle Universal Installer directly from a different directory.

When Oracle Universal Installer is first installed and run, it checks for the JRE path (the location from which it runs), using the location specified in the `oraparam.ini` file's `JRE_LOCATION` parameter. If Oracle Universal Installer cannot find the JRE specified, an error is returned.

Command Line Arguments

Following is the output from the `runInstaller -help` command, which gives you the full list of command line options and their descriptions, as well as command line variables usage:

```
Usage:
runInstaller [-options] [(<CommandLineVariable=Value>)*]
Where options include:
-clusterware oracle.crs,<crs version> Version of Cluster ready services installed.
-crsLocation <Path> Used only for cluster installs, specifies the path to the crs home
location. Specifying this overrides Oracle Clusterware information obtained from central
inventory.
-invPtrLoc <full path of oraInst.loc> Unix only. To point to a different inventory location.
The orainst.loc file contains:

inventory_loc=<location of central inventory>
inst_group=<>
-jreLoc <location> Path where Java Runtime Environment is installed. OUI cannot be run
 without it.
-logLevel <level> To filter log messages that have a lesser priority level than <level>.
 Valid options are: severe, warning, info, config, fine, finer, finest, basic, general,
 detailed, trace. The use of basic, general, detailed, trace is deprecated.
-paramFile <location of file> Specify location of oraparam.ini file to be used by OUI.
-responseFile <Path> Specifies the response file and path to use.
-sourceLoc  <location of products.xml> To specify the shiphome location.
-addLangs To add new languages to an already installed product.
-addNode For adding node(s) to the installation.
-attachHome For attaching homes to the OUI inventory.
-cfs Indicates that the Oracle home specified is on cluster file system
(shared). This is mandatory when '-local' is specified so that Oracle
Universal Installer can register the home appropriately into the inventory.
-clone For making an Oracle Home copy match its current environment.
-debug For getting the debug information from OUI.
-detachHome For detaching homes from the OUI inventory without deleting
inventory directory inside Oracle home.
-enableRollingUpgrade Used in cluster environment, to enable upgrade of a product on a
subset of nodes (on which the product was installed).
-executeSysPrereqs Execute system prerequisite checks and exit.
-force Allowing silent mode installation into a non-empty directory.
```

-help Displays above usage.
-ignorePatchConflicts Ignore all conflicts with existing interim patches during an upgrade. The conflicting interim patches are removed from the home.
-ignoreSysPrereqs For ignoring the results of the system prerequisite checks.
-local Performs the operation on the local node irrespective of the cluster nodes specified.
-printdiskusage Log debug information for disk usage.
-printmemory Log debug information for memory usage.
-printtime Log debug information for time usage.
-removeAllPatches Remove all interim patches from the home
-silent For silent mode operations, the inputs can be a response file or a list of command line variable value pairs.
-updateNodeList For updating node list for this home in the OUI inventory.
-waitforcompletion For windows. setup.exe will wait for completion instead of spawning the java engine and exiting.
-nobackground Do not show background image
-noclusterEnabled No cluster nodes specified.
-noconsole For suppressing display of messages to console. Console is not allocated.
-nowarningonremovefiles To disable the warning message before removal of home directory.
-nowait For windows. Do not wait for user to hit Enter on the console after the task (install etc.) is complete.
-formCluster To install the Oracle clusterware in order to form the cluster.
-remotecp <Path> Unix specific option. Used only for cluster installs, specifies the path to the remote copy program on the local cluster node.
-remoteshell <Path> Unix specific option. Used only for cluster installs, specifies the path to the remote shell program on the local cluster node.

Command Line Variables Usage
Command line variables are specified using <name=value>; for example:
[session: | compName: | compName:version:]variableName="valueOfVariable"]

Session/Installer variables are specified using:
[session:]varName=value
Ex 1: session:ORACLE_HOME="OraHome"
Ex 2: ORACLE_HOME="OraHome"
The lookup order is session:varName then just varName).
The session prefix is used to avoid ambiguity.

Component variables are specified using:
[compInternalName:[Version:]]varName
Ex 1: oracle.comp1:1.0.1:varName="VarValue"
Ex 2: oracle.comp1:varName="VarValue"
Ex 2: oracle.comp1:varName="VarValue"
The lookup order is compInternalName:Version:varName, then compInternalName:varName, then just varName.

Using Oracle Universal Installer Exit Codes

If you are starting and stopping Oracle Universal Installer programmatically (for example, by invoking Oracle Universal Installer using a response file), you may need to consider the exit codes Oracle Universal Installer generates, and perform a particular action depending on the code Oracle Universal Installer returns.

Oracle Universal Installer returns one of the following exit codes:

Code	Description
0	All installations were successful.
1	All installations were successful, but some optional configuration tools failed.
2	Local installations were successful, but some remote operations failed.

Code	Description
3	All installations were successful, but some recommended configuration tools failed.
4	The installation was stopped.
6	The installation was successful after you proceeded by disregarding a few prerequisite checks or warnings.
-1	At least one installation failed.
-2	The installation failed. One or more validation of variables failed.
-3	The attempted installation encountered a prerequisite failure. Some of the optional prerequisites have not been met. See the logs for details.
	You can ignore this code if the prerequisite is optional.

Note that:

- This feature does not work if Oracle Universal Installer is running in "bootstrap" mode. In this case, `setup.exe/runInstaller` just launches the JRE process and returns immediately without waiting for the exit code. Oracle Universal Installer runs in "bootstrap" mode if the following line exists in the `oraparam.ini` file:

  `BOOTSTRAP=TRUE`

- If you exit without installing any products (for example, if you exit from the "Welcome" screen), the exit code is `-1`.

Cloning Considerations

You can copy an existing Oracle home, then configure it for its new environment. This process is called "cloning."

> **Note:** Patching and deinstallation on a cloned Oracle home act the same as a regularly installed Oracle home. You can directly patch a cloned installation.

Invoke Oracle Universal Installer in clone mode using the following command:

`./runInstaller -clone ORACLE_HOME="<target location>" [-responseFile <full path>]`

Use `setup.exe` instead of `runInstaller` for Windows machines. The `-responseFile` parameter is optional. You can supply clone-time parameters on the command line or through the response file named on the command line.

Clone-time activity is logged in the `cloneActions<timestamp>.log` file at installation time.

For more information on cloning see Chapter 6, "Cloning Oracle Software".

> **Note:** Because most cloning is done in silent mode, when cloning an Oracle home onto a "clean" host (one that has no oraInst.loc file), Oracle Universal Installer creates a Central Inventory in the location specified by the `INVENTORY_LOCATION` variable. If this variable is not specified, Oracle Universal Installer creates the Central Inventory in the `<cloned_home>/oraInventory` directory.
>
> After cloning is finished, you must run `oraInstRoot.sh` as `root` to move `oraInventory` to the final, desired location.

About Oracle Universal Installer Log Files

When you install or deinstall products using Oracle Universal Installer, important information about each installation is saved not only in the inventory, but also in a series of log files located in the following directory:

`$ORACLE_HOME/cfgtoollogs`

You can use these log files to troubleshoot installation problems. These files are also crucial for removing and configuring the various software components you install on your Windows or UNIX computer. Oracle Universal Installer displays the name and location of the current session's log file on the Install page. Each installation or configuration utility provides a separate folder containing the installActions<timestamp>.log files inside the $ORACLE_HOME/cfgtoollogs folder.

Many exceptions can possibly occur and consequently appear in a log file, depending on the product, as shown in the following example:

```
globalVarQueries2.1.0.4.1  getGlobalVariable[[.variable =
oracle.assistants.server.launchNETCA]]
[2009-09-07T01:17:46.646+00:00] [OUI] [NOTIFICATION] [] [OUI] [tid: 21]
[ecid: 0000IEI4dFFDScApJ^^Ayf1Ad5uS00000C,0]
[[Query Exception: VariableNotFoundException
Query Exception Class: class oracle.sysman.oii.oiil.OiilQueryException]]
```

You can ignore the exception traces in the logs if the installation has subsequently continued.

Action logs are written on a per-session basis. The installer action log is created each time a new install session is started. What each action does and whether it occurred during installation or deinstallation is logged for every action in the installer action log. Each session is saved in the file, installActionstime_stamp.log, where time_stamp is of the form:

`yyyy-mm-dd_hh-mm-ss[AM|PM].log`

The .err and .out files also use the time stamp in their file names, making it easier to keep track of these files for each session.

Product Removal Logs vs. Action Logs

Note that the logs used to remove products are different from the `installActions<timestamp>.log` file generated during the installation process. The `installActions<timestamp>.log` is easier to read and can be used to view the operations performed during installation.

Exceptions

Many exceptions can possibly occur and consequently appear in a log file, depending on the product, as shown in the following example:

```
globalVarQueries2.1.0.4.1  getGlobalVariable[[.variable =
oracle.assistants.server.launchNETCA]]
[2009-09-07T01:17:46.646+00:00] [OUI] [NOTIFICATION] [] [OUI] [tid: 21]
[ecid: 0000IEI4dFFDScApJ^^Ayf1Ad5uS00000C,0]
[[Query Exception: VariableNotFoundException
Query Exception Class: class oracle.sysman.oii.oiil.OiilQueryException]]
```

You do not need to take any action if the text that follows the exception message appears to ignore the exception.

```
yyyy-mm-dd_hh-mm-ss[AM|PM]
```

5

Installing Cluster Environments

A cluster installation uses Oracle Universal Installer to install software on the nodes of a cluster that are network reachable and bound together by Oracle Clusterware. You can use Oracle Universal Installer to extend the Oracle home of a product installation to include additional nodes on the cluster.

This chapter includes the following sections:

- Oracle Universal Installer and Oracle Real Application Clusters
- General System Installation Requirements for Oracle Real Application Clusters
- Cluster Setup and Pre-installation Configuration Tasks for Oracle Real Application Clusters
- Pre-installation Tasks for Oracle Real Application Clusters on UNIX
- Pre-installation Tasks for Real Application Clusters on Windows
- Configuring Storage for Oracle Clusterware
- Installing Oracle Clusterware on UNIX
- Installing Oracle Clusterware on Windows
- Adding More Nodes to the Cluster for Mass Deployment
- Installing Product Software on a Cluster
- Command Line Options for Cluster Installations (UNIX Only)
- Adding Language Support for Installed Products
- Patchsets and Upgrades
- Post-installation Tasks
- Deinstalling Real Application Clusters Software
- Converting Single-instance Nodes to Real Application Clusters
- Troubleshooting Real Application Clusters / Oracle Clusterware Installation

Oracle Universal Installer and Oracle Real Application Clusters

The Oracle Universal Installer facilitates the installation of Oracle grid infrastructure for a cluster, which includes Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM). You must install Oracle grid infrastructure before you install Oracle Real Application Clusters (Oracle RAC). In most cases, you use the graphical user interface (GUI) provided by the Oracle Universal Installer to install the

software. However, you can also use the Oracle Universal Installer to complete non-interactive (or "silent") installations without using the GUI.

The Oracle inventory maintains records of Oracle software versions and patches. Each installation has a Central Inventory where the Oracle home is registered. Oracle software installations have a local home directory, whose path location is recorded in the Central Inventory. The local inventory directory for each Oracle software installation contains a list of components and applied interim patches associated with that software. Because faulty inventory information can corrupt your Oracle software installation, the Oracle Universal Installer must perform all read and write operations on Oracle inventories. Ensure that you do not modify files in the Central Inventory or the Oracle home inventory.

When you install Oracle grid infrastructure for a cluster or Oracle Real Application Clusters, the Oracle Universal Installer copies the Oracle software onto the node from which you are running it. If your Oracle home is not on a shared file system, the Oracle Universal Installer propagates the software onto the other nodes that you have selected to be part of your Oracle Universal Installer installation session. The Oracle inventory maintains a list of each node that is a member of the Oracle Real Application Clusters database, and lists the paths to each node's Oracle home. This is used to maintain patches and updates for each member node of the Oracle Real Application Clusters database.

When the Oracle Universal Installer installs the Oracle software, Oracle recommends that you select a preconfigured database, or use the Database Configuration Assistant (DBCA) interactively to create your cluster database. You can also manually create your database as described in the *Oracle Database Administrator's Guide*.

Oracle ASM is installed as part of the Oracle grid infrastructure installation. Oracle recommends that you use Oracle ASM. If you are not using Oracle ASM, configure Oracle RAC on a supported storage option.

Oracle Clusterware and Vendor Clusterware Installations

The Oracle Universal Installer installs Oracle Clusterware on each node on which the Oracle Universal Installer detects that third-party vendor clusterware is present. If third-party vendor clusterware is not present, you must use the Oracle Universal Installer to enter the nodes on which you want the Oracle Clusterware to be installed. The Oracle Clusterware home can be either shared by all nodes, or private to each node, depending on your responses when you run the Oracle Universal Installer.

When third-party vendor clusterware is present, Oracle Clusterware can interact with the third-party vendor clusterware. In using third-party vendor clusterware, note that Oracle Clusterware can integrate with third-party vendor clusterware for all operating systems except Linux and Windows. You need to have Oracle Cluster Registry and Voting Disks shared across all nodes.

Installed Oracle Real Application Clusters Components

All instances in an Oracle Real Application Clusters environment share the control file, server parameter file, redo log files, and all data files. These files reside on a shared cluster file system or on shared disks. Either of these types of file configurations are accessed by all the cluster database instances. Each instance also has its own set of redo log files. During failures, shared access to redo log files enables surviving instances to perform recovery.

General System Installation Requirements for Oracle Real Application Clusters

Each node that is going to be part of your Oracle Real Application Clusters installation must meet the following hardware and software requirements. You will perform step-by-step tasks for hardware and software verification for the platform-specific pre-installation procedures.

Hardware Requirements for Oracle Real Application Clusters Setup

Each node in a cluster requires the following hardware:

■ External shared disks for storing the Oracle Clusterware files.

See the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)* for information on the available disk configuration options. Review these options before you decide which storage option to use in your Oracle Real Application Clusters environment.

■ One private Internet protocol (IP) address for each node to serve as the private interconnect. The following must be true for each private IP address:

– It must be separate from the public network.

– It must be accessible on the same network interface on each node.

– It must have a unique address on each node.

The private interconnect is used for inter-node communication by both Oracle Clusterware and Oracle Real Application Clusters. If the private address is available from a network name server (DNS), you can use that name. Otherwise, the private IP address must be available in each node's `/etc/hosts` file on UNIX and `%SystemRoot%\system32\drivers\etc\hosts` on Windows.

During Oracle Clusterware installation, the information you enter as the private IP address determines which private interconnects are used by Oracle Real Application Clusters database instances.

■ One public IP address for each node to be used as the Virtual IP address for client connections and for connection failover.

This public Virtual IP address (VIP) must be associated with the same interface name on every node that is part of your cluster. Additionally, the IP addresses that you use for all of the nodes that are part of a cluster must be from the same subnet. If you have a domain name server (DNS), register the host names for the VIP with DNS. The Virtual IP address should not be in use at the time of the installation, because this is a Virtual IP address that Oracle manages.

■ One public fixed host name address for each node, typically assigned by the system administrator during operating system installation. If you have a DNS, register both the fixed IP and the VIP address with DNS. If you do not have DNS, you must make sure that both public IP addresses are in the node host file.

Software Requirements for Oracle Real Application Clusters Setup

Each node in a cluster requires a supported interconnect software protocol to support Cache Fusion, and to support Oracle Clusterware polling. Your interconnect must be certified by Oracle for your platform. You should also have a Web browser, both to enable Oracle Enterprise Manager, and to view online documentation. For Oracle Database 11*g* requirements, Oracle Clusterware provides the same functionality as third-party vendor clusterware. Using Oracle Clusterware also reduces installation

and support complications. However, you may require third-party vendor clusterware if you use a non-ethernet interconnect, or if you have deployed clusterware-dependent applications on the same cluster where you deploy Oracle Real Application Clusters.

Cluster Setup and Pre-installation Configuration Tasks for Oracle Real Application Clusters

Before installing Oracle Real Application Clusters, perform the following procedures:

1. Ensure that you have a certified combination of operating system and Oracle software version by referring to the OracleMetaLink certification information, which is located at the following Web site:

   `http://metalink.oracle.com`

   Click **Certify & Availability**, and select **1.View Certifications by Product**.

   > **Note:** The layout of the OracleMetaLink site and the site's certification policies are subject to change.

2. Configure a high-speed interconnect that uses a private network. Some platforms support automatic failover to an additional interconnect.

3. Determine the storage option for your system and configure the shared disk.

4. Ensure that the directory structure is similar in all nodes.

5. Ensure that all nodes in the cluster have the same time zone settings.

6. Install the operating system patches for the Oracle software. For more information, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Pre-installation Tasks for Oracle Real Application Clusters on UNIX

You need to complete the following pre-installation tasks before you start the Oracle Universal Installer:

Logging In to the System as root

Before you install the Oracle software, you must complete several tasks as the root user. To log in as the root user, complete one of the following procedures:

- If you are installing the software from an X Window System workstation or X terminal, then:

  1. Start a local terminal session, for example, an X terminal (xterm).

  2. If you are not installing the software on the local system, enter the following command to enable remote hosts to display X applications on the local X server:

     `$ xhost +`

  3. If you are not installing the software on the local system, use the `ssh`, `rlogin`, or `telnet` command to connect to the system where you want to install the software:

     `$ telnet remote_host`

4. If you are not logged in as the root user, enter the following command to switch user to `root`:

```
$ su - root
password:
#
```

■ If you are installing the software from a PC or other system with X server software installed:

1. Start the X server software.

2. Configure the security settings of the X server software to permit remote hosts to display X applications on the local system.

3. Connect to the remote system where you want to install the software and start a terminal session on that system, for example, an X terminal (xterm).

4. If you are not logged in as the root user on the remote system, enter the following command to switch user to `root`:

```
$ su - root
password:
#
```

Creating Required UNIX Groups and Users

Depending on whether it is the first time an Oracle software product is being installed on a system and on the products that you are installing, you may need to create several UNIX groups and a UNIX user account. The following UNIX group and users are required for all installations:

■ The Oracle Inventory group (`oinstall`)

You must create this group the first time you install Oracle software on the system. The usual name chosen for this group is `oinstall`. This group owns the Oracle inventory, which is a catalog of all Oracle software installed on the system.

> **Note:** If Oracle software is already installed on the system, the existing Oracle Inventory group must be the primary group of the UNIX user that you use to install new Oracle software. The following sections describe how to identify an existing Oracle Inventory group.

■ The Oracle software owner user (`oracle`)

You must create this user the first time you install Oracle software on the system. This user owns all of the software installed during the installation. The usual name chosen for this user is `oracle`. This user must have the Oracle Inventory group as its primary group.

You need to create other groups and users in addition to the ones created above. For more information on creating the required groups and users, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Creating Identical Users and Groups on Other Cluster Nodes

The Oracle software users and groups must exist and be identical on all cluster nodes. To create these identical users and groups, you must identify the user ID and group IDs assigned to them on the node where you created them, then create the user and groups with the same name and ID on the other cluster nodes. You must create

identical users and groups on other cluster nodes only if you are using local users and groups. If you are using users and groups defined in a directory service such as NIS, they are already identical on each cluster node.

Identifying the User and Group IDs To determine the user ID (UID) and the group IDs (GID) for the groups, follow these steps:

1. Enter following command:

   ```
   # id oracle
   ```

   The output from this command is similar to the following:

   ```
   uid=440(oracle) gid=200(oinstall) groups=201(dba),202(oper)
   ```

2. From the output, identify the user identity (UID) for the Oracle user and the group identities (GIDs) for the groups to which it belongs.

Creating the User and Groups on the Other Cluster Nodes To create the user and groups on the other cluster nodes, repeat the following procedure on each node:

1. Log in to the cluster node in which you want to create the user and groups as `root`.

2. Enter commands as per the syntax to create the respective groups. Use the `-g` option to specify the correct GID for each group:

   ```
   # /usr/sbin/groupadd -g <group_id> <group_name>
   ```

Configuring SSH on all Cluster Nodes

Before you install and use Oracle Real Application Clusters, you must configure secure shell (SSH) for the `oracle` user on all cluster nodes. Oracle Universal Installer uses the `ssh` and `scp` commands during installation to run remote commands on and copy files to the other cluster nodes. You must configure SSH so that these commands do not prompt for a password. SSH is also used by the configuration assistants, Enterprise Manager, and when adding nodes to the cluster.

You can configure SSH from the Oracle Universal Installer (OUI) interface during installation for the user account running the installation. The automatic configuration creates passwordless SSH connectivity between all cluster member nodes. You can use an SSH Connectivity button available in the Cluster Node information page for this purpose.

To enable the script to run, you must remove `stty` commands from the profiles of any Oracle software installation owners, and remove other security measures triggered during a login and that generate messages to the terminal. These messages, mail checks, and other displays prevent Oracle software installation owners from using the SSH configuration script that is built into the Oracle Universal Installer. If they are not disabled, you must manually configure SSH before running an installation.

On Linux systems, to enable Oracle Universal Installer to use the `ssh` and `scp` commands without being prompted for a pass phrase, you must have user equivalency in the cluster. User equivalency exists in a cluster when the following occurs on all nodes in the cluster:

- A given user has the same user name, user ID (UID), and password.

- A given user belongs to the same groups.

- A given group has the same group ID (GID).

> **Note:** This section describes how to configure OpenSSH version 3. If SSH is not available, then Oracle Universal Installer attempts to use `rsh` and `rcp` instead. However, these services are disabled by default on most Linux systems.

Configuring SSH on Cluster Member Nodes To configure SSH, complete the following steps on each cluster node:

1. Log in as the Oracle user.

2. If necessary, create the `.ssh` directory in the Oracle user's home directory and set the correct permissions for it:

   ```
   $ mkdir ~/.ssh
   $ chmod 700 ~/.ssh
   ```

3. Enter the following commands to generate an RSA key for version 2 of the SSH protocol:

   ```
   $ /usr/bin/ssh-keygen -t rsa
   ```

   At the prompts:

   - Accept the default location for the key file.

   - Enter and confirm a different pass phrase from the Oracle user's password.

   This command writes the public key to the `~/.ssh/id_rsa.pub` file and the private key to the `~/.ssh/id_rsa` file. Never distribute the private key to anyone.

4. Enter the following command to generate a DSA key for version 2 of the SSH protocol:

   ```
   $ /usr/bin/ssh-keygen -t dsa
   ```

   At the prompts:

   - Accept the default location for the key file.

   - Enter and confirm a pass phrase that is different from the Oracle user's password.

   This command writes the public key to the `~/.ssh/id_dsa.pub` file and the private key to the `~/.ssh/id_dsa` file. Never distribute the private key to anyone.

5. Copy the contents of the `~/.ssh/id_rsa.pub` and `~/.ssh/id_dsa.pub` files to the `~/.ssh/authorized_keys` file on this node and to the same file on all other cluster nodes.

   > **Note:** The `~/.ssh/authorized_keys` file on every node must contain the contents from all of the `~/.ssh/id_rsa.pub` and `~/.ssh/id_dsa.pub` files that you generated on all cluster nodes.

6. Change the permissions on the `~/.ssh/authorized_keys` file on all cluster nodes:

   ```
   $ chmod 600 ~/.ssh/authorized_keys
   ```

At this point, if you use ssh to log in to or run a command on another node, you are prompted for the pass phrase that you specified when you created the DSA key.

Enabling SSH User Equivalency on Cluster Member Nodes   To enable Oracle Universal Installer to use the ssh and scp commands without being prompted for a pass phrase, follow these steps:

1. On the system where you want to run Oracle Universal Installer, log in as the Oracle user.

2. Enter the following commands:

    ```
    $ exec /usr/bin/ssh-agent $SHELL
    $ /usr/bin/ssh-add
    ```

3. At the prompts, enter the pass phrase for each key that you generated.

    If you have configured SSH correctly, then you can use the ssh or scp commands without being prompted for a password or a pass phrase.

4. To test the SSH configuration, enter the following commands from the same terminal session, testing the configuration of each cluster node:

    ```
    $ ssh nodename1 date
    $ ssh nodename2 date
    .
    .
    ```

    These commands should display the date set on each node. If any node prompts for a password or pass phrase, verify that the ~/.ssh/authorized_keys file on that node contains the correct public keys.

    > **Note:**   The first time you use SSH to connect to a node from a particular system, you might see a message stating that the authenticity of the host could not be established. Enter yes at the prompt to continue. You should not see this message again when you connect from this system to that node.
    >
    > If you see any other messages or text, apart from the date, the installation can fail. Make any changes required to ensure that only the date is displayed when you enter these commands.
    >
    > You should ensure that any parts of login scripts that generate any output, or ask any questions, are modified so that they only act when the shell is an interactive shell.

5. To ensure that X11 forwarding does not cause the installation to fail, create a user-level SSH client configuration file for the Oracle software owner user, as follows:

    a. Using any text editor, edit or create the ~oracle/.ssh/config file.

    b. Make sure that the ForwardX11 attribute is set to no. For example:

        ```
        Host *
        ForwardX11 no
        ```

6. You must run Oracle Universal Installer from this session or remember to repeat steps 2 and 3 before you start Oracle Universal Installer from a different terminal session.

Preventing Oracle Clusterware Installation Errors Caused by stty Commands During an Oracle Clusterware installation, the Oracle Universal Installer uses SSH to run commands and copy files to the other nodes. During the installation, hidden files on the system (for example, .bashrc or .cshrc) can cause installation errors if they contain stty commands.

To avoid this problem, Oracle recommends that you modify these files to suppress all output on STDERR, as in the following examples:

- Bourne, Bash, or Korn shell:

  ```
  if [ -t 0 ]; then
  stty intr ^C
  fi
  ```

- C shell:

  ```
  test -t 0
  if ($status == 0) then
  stty intr ^C
  endif
  ```

Configuring the Oracle User Environment

You run Oracle Universal Installer from the oracle account. However, before you start Oracle Universal Installer you must configure the environment of the oracle user. For more information on configuring the environment, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Checking the Hardware Requirements

Each system must meet certain minimum hardware requirements. For more information on hardware requirements, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Checking the Network Requirements

There are some minimum networking hardware and Internet protocol (IP) address requirements. Check that you have the networking hardware and Internet protocol (IP) address required for an Oracle Real Application Clusters installation.

Network Hardware Requirements

Each node in the cluster must meet the following requirements:

- Each node must have at least two network adapters: one for the public network interface and one for the private network interface (the interconnect).

- The interface names associated with the network adapters for each network must be the same on all nodes.

- For increased reliability, you can configure redundant public and private network adapters for each node.

- For the public network, each network adapter must support TCP/IP.

- For the private network, the interconnect must support the user datagram protocol (UDP) using high-speed network adapters and switches that support TCP/IP (Gigabit Ethernet or better recommended).

> **Note:** UDP is the default interconnect protocol for Real Application Clusters and TCP is the interconnect protocol for Oracle Clusterware. Token-Ring is not supported for the interconnect.

- For the private network, the end points of all designated interconnect interfaces must be completely reachable on the network.

Network Parameter Requirements

If you are using NFS, then you must set the values for the NFS buffer size parameters rsize and wsize to at least 16384. Oracle recommends that you use the value 32768.

IP Address Requirements

Before starting the installation, you must identify or obtain the following IP addresses for each node:

- An IP address and an associated host name registered in the domain name service (DNS) for each public network interface.

- One unused virtual IP address and an associated virtual host name registered in DNS, or resolved in the host file, or both, that you will configure for the primary public network interface.

  The virtual IP address must be in the same subnet as the associated public interface. After installation, you can configure clients to use the virtual host name or IP address. If a node fails, then its virtual IP address fails over to another node.

Node Time Requirements

Before starting the installation, ensure that each member node of the cluster is set as closely as possible to the same date and time. Oracle strongly recommends using the Network Time Protocol feature of most operating systems for this purpose, with all nodes using the same reference Network Time Protocol server.

Checking Software Requirements

The Oracle Universal Installer performs checks on your system to verify that it meets the requirements listed for your platform. To ensure that these checks pass, verify the requirements before you start the Installer.

Depending on the Oracle products that you intend to install, verify that the required software is installed on the system. For more information on software requirements, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Configuring Kernel Parameters

Each cluster node must meet the recommended kernel values. For information on verifying and setting the kernel parameters, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Identifying Required Software Directories

You must identify or create the following directories for the Oracle software as follows:

- Oracle Base Directory
- Oracle Inventory Directory

- Oracle Clusterware Home Directory

- Oracle Home Directory

The following subsections describe the requirements for these directories.

Oracle Base Directory

The Oracle base directory acts as a top-level directory for Oracle software installations. It is analogous to the `C:\Oracle` directory used for Oracle software installations on Windows systems. On UNIX systems, the Optimal Flexible Architecture (OFA) guidelines recommend that you use a path similar to the following for the Oracle base directory:

```
/mount_point/app/oracle_sw_owner
```

In this example:

- `mount_point` is the mount point directory for the file system that will contain the Oracle software.

    The examples in this guide use `/u01` for the mount point directory. However, you could choose another mount point directory, such as `/oracle or /opt/oracle` for example.

- `oracle_sw_owner` is the operating system user name of the Oracle software owner, for example `oracle`.

You can use the same Oracle base directory for more than one installation, or you can create separate Oracle base directories for different installations. If different operating system users install Oracle software on the same system, each user must create a separate Oracle base directory. The following example Oracle base directories could all exist on the same system:

```
/u01/app/oracle
```

```
/u01/app/orauser
```

```
/opt/oracle/app/oracle
```

Regardless of whether you create an Oracle base directory or decide to use an existing one, you must set the `ORACLE_BASE` environment variable to specify the full path to this directory.

> **Note:** The Oracle base directory can be on a local file system or on an NFS file system on a certified NAS device. Do not create the Oracle base directory on an OCFS version 1 file system.

Oracle Inventory Directory

The Oracle Inventory directory (oraInventory) stores an inventory of all software installed on the system. It is required by, and shared by, all Oracle software installations on a single system. The first time you install Oracle software on a system, Oracle Universal Installer prompts you to specify the path to this directory. If you are installing the software on a local file system, Oracle recommends that you choose the following path:

```
oracle_base/../oraInventory
```

If the Oracle base directory is on a cluster file system, or on an NFS file system on a NAS device, you must specify a path for the Oracle Inventory directory on a local file system. The Oracle base directory must be on a local file system to enable all of the nodes to have separate inventories.

Oracle Universal Installer creates the directory that you specify, and sets the correct owner, group, and permissions for it. You do not need to create it.

> **Note:** All Oracle software installations rely on the Oracle base directory. Make sure that you back it up regularly.
>
> Do not delete the Oracle base directory unless you have completely removed all Oracle software from the system.

Oracle Clusterware Home Directory

The Oracle Clusterware home directory is the directory where you choose to install the software for Oracle Clusterware. You must install Oracle Clusterware in a separate home directory. When you run Oracle Universal Installer, it prompts you to specify the path to this directory, as well as a name that identifies it. Oracle recommends that you specify a path similar to the following for the Oracle Clusterware home directory:

```
/u01/crs/oracle/product/11.2.0/app
```

> **Note:** Because you must change the permissions of all of the parent directories of the Oracle Clusterware home directory after installing the software to grant write access only to the root user, the Oracle Clusterware home directory must not be a subdirectory of the Oracle base directory.

Oracle Home Directory

The Oracle home directory is the directory where you choose to install the software for a particular Oracle product. You must install different Oracle products, or different releases of the same Oracle product, in separate Oracle home directories. When you run Oracle Universal Installer, it prompts you to specify the path to this directory, as well as a name that identifies it. The directory that you specify must be a subdirectory of the Oracle base directory. Oracle recommends that you specify a path similar to the following for the Oracle home directory:

```
oracle_base/product/11.2.0/db_1
```

Oracle Universal Installer creates the directory path that you specify under the Oracle base directory. It also sets the correct owner, group, and permissions on it. You do not need to create this directory.

Identifying or Creating an Oracle Base Directory

Before starting the installation, you must either identify an existing Oracle base directory or, if required, create one. For more information on identifying or creating an Oracle Base Directory, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Creating the Clusterware Home Directory

Oracle Universal Installer creates the Oracle Clusterware home directory for you. Ensure before you start the installation that you provide sufficient disk space on a file system for the Oracle Clusterware directory, and the parent directory of the Oracle Clusterware directory space is writable by the Oracle user.

To create the Oracle Clusterware home directory and specify the correct owner, group, and permissions for it, follow these steps:

1. Enter commands similar to the following to create the recommended subdirectories in the mount point directory that you identified, and set the appropriate owner, group, and permissions on them:

   ```
   # mkdir -p /mount_point/crs/oracle_sw_owner/product/11.2.0/crs
   # chown -R root:oinstall /mount_point/crs
   # chmod -R 775 /mount_point/crs/oracle_sw_owner
   ```

   If the mount point you identified is `/u01`, the recommended Oracle Clusterware home directory path is as follows:

   ```
   /u01/crs/oracle/product/11.2.0/crs
   ```

2. If necessary, repeat the commands listed in the previous step to create the same directory on the other nodes in the cluster.

3. Enter commands similar to the following to set the `ORACLE_BASE` and `ORACLE_HOME` environment variables in preparation for the Oracle Clusterware installation:

   - Bourne, Bash, or Korn shell:

     ```
     $ ORACLE_BASE=/u01/app/oracle
     $ ORACLE_HOME=/u01/crs/oracle/product/11.2.0/crs
     $ export ORACLE_BASE
     $ export ORACLE_HOME
     ```

   - C shell:

     ```
     % setenv ORACLE_BASE /u01/app/oracle
     % setenv ORACLE_HOME /u01/crs/oracle/product/11.2.0/crs
     ```

4. Enter the following commands to ensure that the `TNS_ADMIN` environment variable is not set:

   - Bourne, Bash, or Korn shell:

     ```
     $ unset TNS_ADMIN
     ```

   - C shell:

     ```
     % unsetenv TNS_ADMIN
     ```

5. To verify that the environment has been set correctly, enter the following commands:

   ```
   $ umask
   $ env | more
   ```

   Verify that the `umask` command displays a value of 22, 022, or 0022, and the environment variables that you set in this section have the correct values.

Pre-installation Tasks for Real Application Clusters on Windows

You need to complete the following pre-installation tasks before you start the Oracle Universal Installer to install Oracle Clusterware and Oracle Real Application Clusters on Microsoft Windows systems.

Checking Hardware and Software Certification

Oracle recommends you to review the certification matrix on the OracleMetaLink Web site for the most up-to-date list of certified hardware platforms and operating system versions for Oracle software. This Web site also provides patches and workaround information for bugs. For more information on accessing the OracleMetaLink site, see "Download and Install Patches" on page 5-23.

Checking the Hardware Requirements

Each system must meet minimum hardware requirements. For more information on hardware requirements and the steps involved to check them, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Hard Disk Space Requirements

Each system must meet certain hard disk space requirements. For more information on hard disk space requirements, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Checking the Software Requirements

Each system must meet minimum software requirements. For more information on software requirements and the steps involved to check them, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Checking the Network Requirements

There are some minimum networking hardware and Internet protocol (IP) address requirements. Check that you have the networking hardware and Internet protocol (IP) address required for an Oracle Real Application Clusters installation. For more information on the steps involved in checking these requirements, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Network Hardware Requirements

Each node in the cluster must meet the following requirements:

- Each node must have at least two network adapters: one for the public network interface and one for the private network interface (the interconnect).

- The private and public network interface names must be different from each other and cannot contain any multibyte language characters. The names are case-sensitive.

- The private network interface name must be the same on all nodes.

- The public network interface name must be the same on all nodes.

- The public interface must be listed first in the ipconfig list.

- Oracle supports the TCP/IP protocol for the public and private networks.

IP Address Requirements

Before starting the installation, you must identify or obtain the following IP addresses for each node:

- An IP address and an associated host name registered in the domain name service (DNS) for each public network interface.

- One unused virtual IP address and an associated virtual host name registered in DNS that you will configure for the primary public network interface.

  The virtual IP address must be in the same subnet as the associated public interface. After installation, you can configure clients to use the virtual host name or IP address. If a node fails, then its virtual IP address fails over to another node.

- A private IP address and optional host name for each private interface.

  Oracle recommends that you use private network IP addresses for the private interfaces; for example: 10.*.*.* or 192.168.*.*. You can use the `%SystemRoot%\system32\drivers\etc\hosts` file on each node to associate private host names with private IP addresses.

Checking Individual Component Requirements

Depending on the Oracle products that you intend to install, verify the required components for the Oracle software you are installing and ensure they are installed on the system. You can find more information on the components in the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Configuring Storage for Oracle Clusterware

You need to perform certain storage configuration tasks before you start Oracle Universal Installer. For detailed information on the storage configuration tasks, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)*.

Installing Oracle Clusterware on UNIX

This section provides you with information about how to use the Oracle Universal Installer to install Oracle Clusterware on UNIX.

Running the Oracle Universal Installer to Install Oracle Clusterware

Complete the following steps to install Oracle Clusterware on your cluster. At any time during installation, if you have a question about what you are being asked to do, click **Help** in the Oracle Universal Installer page.

1. Start the `runInstaller` command with the `-formCluster` option from the main directory on the Oracle Clusterware 11*g* Release 2 (11.2) CD-ROM. This is a separate CD that contains the Oracle Clusterware software. When the Oracle Universal Installer displays the **Welcome** page, click **Next**.

2. Oracle Universal Installer displays the **Installation Prerequisite Checks** page. This screen shows the name, type, and status for all prerequisite checks designed for the installation. Click **Next.**

3. Oracle Universal Installer displays the **Specify Cluster Nodes for Node Addition** page. Specify public, private, virtual IPs in this page and click **Next**.

4. Oracle Universal Installer displays the **Specify Local Host Name** page. Specify at least one public and one private interface in this page and click **Next**.

5. Specify the Oracle Cluster Registry location option to provide a redundant OCR location and click **Next.**

6. Specify the VD (Voting Disk Location) location option to provide redundant VD locations and click **Next.**

7. Oracle Universal Installer displays the **Summary** page. Click **Install** and proceed.

8. After the installation is done, Oracle Universal Installer prompts you to run `orainstRoot.sh` and `root.sh`. You need to run these scripts and click **OK**.

9. After you run `root.sh` on all of the nodes, the Oracle Universal Installer runs the Oracle Notification Server Configuration Assistant and Oracle Private Interconnect Configuration Assistant. These assistants run without user intervention.

10. Oracle Universal Installer displays the **End of Installation** page. Click **Exit.**

Running the Oracle Universal Installer in Silent Mode to Install Oracle Clusterware

You can run the Oracle Universal Installer in silent mode to install Oracle Clusterware by entering the following command:

```
./runInstaller -silent -responseFile <response file path>
```

For Oracle Clusterware, you need to use the `-formCluster` flag. You can pass session variables like nodelist, ocr, voting disk location, and so forth from the command line for silent installation.

Oracle Clusterware Background Processes

The following processes must be running in your environment after the Oracle Clusterware installation for Oracle Clusterware to function:

- `oprocd` — Process monitor for the cluster.

- `evmd` — Event manager daemon that starts the `racgevt` process to manage call-outs.

- `ocssd` — Manages cluster node membership and runs as the Oracle user; failure of this process results in node restart.

- `crsd` — Performs high availability recovery and management operations such as maintaining the OCR. Also manages application resources and runs as the root user and restarts automatically upon failure.

Installing Oracle Clusterware on Windows

This section provides you with information about how to use the Oracle Universal Installer to install Oracle Clusterware on Windows.

Running the Oracle Universal Installer to Install Oracle Clusterware

Complete the following steps to install Oracle Clusterware on your cluster. At any time during installation, if you have a question about what you are being asked to do, click **Help** in the Oracle Universal Installer page.

1. Log in to Windows with Administrative privileges and run the `setup.exe` command on the Oracle Clusterware media. This opens the Oracle Universal Installer **Welcome** page.

2. After you click **Next** in the **Welcome** page, the **Specify File Locations** page enables you to accept the displayed path name for the Oracle Clusterware products or select a different one. You can also accept default directory and path name for the location of your Clusterware home or browse for an alternate directory and destination. You must select a destination that exists on each cluster node that is part of this installation. Click **Next** to confirm your choices and proceed to the **Language Selection** page.

3. Select the language or languages for your Oracle Clusterware installation in the **Language Selection** page, then click **Next** for the **Cluster Configuration** page.

4. The **Cluster Configuration** page contains predefined node information if the Oracle Universal Installer detects that your system has the Oracle 10*g* Release 2 Clusterware. Otherwise, the Oracle Universal Installer displays the **Cluster Configuration** page without predefined node information.

Provide your own cluster name if you do not want to use the name provided by the Oracle Universal Installer. Note that the selected cluster name must be globally unique throughout the enterprise, and its allowable character set is the same as that for host names; that is, underscores (_), hyphens (-), and single-byte alphanumeric characters (a to z, A to Z, and 0 to 9).

Enter a public and a private host name for each node. Neither the public nor the private name should have a domain qualifier. When you enter the public host name, use the primary host name of each node; that is, the name displayed by the `hostname` command. The private node refers to an address that is only accessible by the other nodes in this cluster, and which Oracle uses for Cache Fusion processing. You can enter either a private host name or a private IP address for each node.

Click **Next** after you have entered the cluster configuration information. This saves your entries and opens the **Specify Network Interface Usage** page.

5. In the **Specify Network Interface Usage** page, the Oracle Universal Installer displays a list of cluster-wide interfaces. Use the drop-down menus on this page to classify each interface as Public, Private, or Do Not Use. The default setting for each interface is Do Not Use. You must classify at least one interconnect as Public and one as Private. Click **Next** when you have made your selections to open the **Select Disk Formatting Options** page.

6. In the **Select Disk Formatting Options** page, you indicate what OCFS you want the Oracle Universal Installer to format for you.

The formatting options are as follows:

- Format two logical drives for data and software storage. In this case, the Oracle Universal Installer creates two cluster file systems, one for the database files and one for the Oracle home. The OCR and voting disk are created in the data file directory.

- Format one logical drive for software storage. In this case, the Oracle Universal Installer creates one cluster file system for the Oracle home and requires two additional partitions, one for the OCR and one for the voting disk. If you select this option, you must have already configured the partitions before proceeding with the installation. These partitions are stamped with "ocrcfg" for the OCR and "votedsk" for the voting disk.

- Format one logical drive for data file storage. In this case, the Oracle Universal Installer creates one cluster file system for the database files which is also used to store the OCR and voting disk.

- Do not format any logical drives. In this case, the Oracle Universal Installer requires two partitions: one for the OCR and one for the voting disk. If you select this option, you must have already configured the partitions before proceeding with the installation. These partitions are stamped with "ocrcfg" for the OCR and "votedsk" for the voting disk.

Click **Next** after making your selection in the **Select Disk Formatting Options** page. The next page that the Oracle Universal Installer displays depends on your selection on the **Select Disk Formatting Options** page, as described in the following step.

7. If you selected the "Format two logical drives for data and software storage" option on the **Select Disk Formatting Options** page, you must complete the **Select Software Storage Drive** page and the **Select Data Storage Drive** page as described in Steps a and b, respectively.

   If you selected the "Format one logical drive for software storage" option on the **Select Disk Formatting Options** page, you must complete the **Select Software Storage Drive** page, the **Disk Configuration - Oracle Cluster Registry (OCR)** page, and the **Disk Configuration - Voting Disk** page as described in Steps a, c, and d, respectively.

   If you selected the "Format one logical drive for data file storage" option on the **Select Disk Formatting Options** page, you must complete the **Select Data Storage Drive** page as described in Step b.

   If you selected the "Do not format any logical drives" option on the **Select Disk Formatting Options** page, you must complete the **Disk Configuration - Oracle Cluster Registry (OCR)** page and the **Disk Configuration - Voting Disk** page as described in Steps c and d, respectively.

   a. On the **Select Software Storage Drive** page, choose a shared drive to see a list of available partitions on that drive. Choose a partition with sufficient space to hold your Oracle home and select the partition's disk number and partition number from the list. Click **Next** to proceed.

   b. On the **Select Data Storage Drive** page, choose a shared drive to see a list of available partitions on that drive. Choose a partition with sufficient space to hold your database files and select the partition's disk number and partition number from the list. Click **Next** to proceed.

   c. On the **Disk Configuration - Oracle Cluster Registry (OCR)** page, choose a partition with sufficient space to hold your OCR and select the partition's disk number and partition number from the list. Click **Next** to proceed.

   d. On the **Disk Configuration - Voting Disk** page, choose a partition with sufficient space to hold your voting disk and select the partition's disk number and partition number from the list. Click **Next** to proceed.

   > **Note:** The Oracle Universal Installer pages described in this step displays logical drives from which you must make your selections. To be valid for selection, a logical drive must be located on a disk without a primary partition.

8. After you click **Next**, the Oracle Universal Installer checks whether the remote inventories are set. If they are not set, the Oracle Universal Installer sets up the remote inventories by setting registry keys. The Oracle Universal Installer also verifies the permissions to enable writing to the inventory directories on the

remote nodes. After completing these actions, the Oracle Universal Installer displays a **Summary** page that shows the cluster node information along with the space requirements and availability. Verify the installation that the Oracle Universal Installer is about to perform and click **Finish**.

9. When you click **Finish**, the Oracle Universal Installer installs the OCFS and Oracle Clusterware software on the local node and validates the installation again. The Oracle Universal Installer also creates any required OCFS file systems. After validating the installation, the Oracle Universal Installer completes the Oracle Clusterware software installation and configuration on the remote nodes.

Running the Oracle Universal Installer in Silent Mode to Install Oracle Clusterware

You can run the Oracle Universal Installer in silent mode to install Oracle Clusterware by entering the following command:

```
./setup.exe -silent -responseFile <response file path>
```

For Oracle Clusterware, you need to use the `-formCluster` flag. You can pass session variables like nodelist, ocr, voting disk location, and so forth from the command line for silent installation.

Oracle Clusterware Background Processes

The following processes must be running in your environment after the Oracle Clusterware installation for Oracle Clusterware to function:

- `oprocd` — Process monitor for the cluster.

- `evmd` — Event manager daemon that starts the `racgevt` process to manage call-outs.

- `ocssd` — Manages cluster node membership and runs as the Oracle user; failure of this process results in node restart.

- `crsd` — Performs high availability recovery and management operations such as maintaining the OCR. Also manages application resources and runs as the root user and restarts automatically upon failure.

Adding More Nodes to the Cluster for Mass Deployment

If you want to add more nodes to the cluster, complete the following procedure:

1. Perform a single-node cluster Oracle Clusterware installation on node1.

2. Execute the prerequisite checks required on the remote nodes in order to add the nodes:

```
$CRS_HOME/bin/cluvfy stage -pre nodeadd -n <new nodes separated by commas>
-verbose
```

3. Go to the `$CRS_HOME/oui/bin` directory. The variable `$CRS_HOME` represents the Oracle Clusterware home directory of the successfully installed Oracle Clusterware software.

4. Run the following command:

```
./addNode.sh -silent "CLUSTER_NEW_NODES=<comma-separated new nodes>"
"CLUSTER_NEW_VIRTUAL_HOSTNAMES=<comma-separated new vips>"
```

`addNode` is not supported in GUI mode in version 11.2.

5. If prompted to do so, run the `orainstroot.sh` script on the new nodes.

6. When prompted, run the `root.sh` script on each of the new nodes.

Installing Product Software on a Cluster

After you have installed clusterware and completed pre-installation tasks, you are ready to install the cluster-ready product on the cluster. To do this, you can use Oracle Universal Installer in interactive mode, command line mode, or silent mode using a response file.

> **See Also:** See "Modes of Installation" on page 4-5 for more information on the different modes.

Cluster Detection

When Oracle Universal Installer detects that the destination you entered on the **File Locations** page is part of a cluster, one of two dialogs appear:

- If the Oracle home name entered is new, the **Specify Hardware Installation Mode** page appears. Use this page to select between cluster and non-cluster installation:

  - **Cluster installation:** Select the nodes for your installation from the list of nodes on which clusterware has been installed.

  - **Non-cluster installation:** Install on a single-node only, even though that node is part of a cluster.

- If the Oracle home name entered is a pre-existing home, the **Selected Nodes** page appears. This is an information-only page that displays the nodes associated with the Oracle home. The installation is performed on all selected nodes. If any of the selected nodes are down, the installation is not allowed to proceed.

> **Note:** In Oracle Universal Installer, click **Installed Products** to display the Inventory panel, which lists all installed Oracle homes. Clustered Oracle homes, which are homes installed on multiple cluster nodes, have an attribute called Cluster Nodes, which lists all the nodes associated with that home.

Availability Checking

When the user clicks **Next** on the **Specify Hardware Installation Mode** page or **Selected Nodes** page, Oracle Universal Installer performs the following availability checks on the nodes:

- **Network reachability:** Tests whether the remote cluster nodes are up and running.

- **Network configuration:** Tests whether the remote cluster nodes are properly network-configured.

- **Inventory setup:** Tests whether the inventory is set up on the remote nodes. If not set up on UNIX, Oracle Universal Installer prompts you to run root scripts on the remote nodes to set up the inventory. On Windows, Oracle Universal Installer sets up the inventory location in the Windows registry.

- **Inventory permission:** Tests whether you can write to the Central Inventory location.

- **Oracle home permission:** Tests whether you can write to the Oracle home.

If all selected nodes are available, the installation process continues to the next step. If a node (or nodes) is not available, the page redisplays with a **Status** column indicating the results of the check. You must fix the problem or choose another set of nodes in order to proceed.

If any of the remote nodes is unreachable, check if those nodes are up and running and if they are properly network-configured. If either the inventory or the Oracle home is not writable, check for the appropriate permissions on the remote nodes for these directories.

Cluster Installation

In a typical cluster installation, when clusterware is present, Universal Installer installs the Oracle software onto the node on which Oracle Universal Installer is running, then propagates the Oracle home from the local node installation to the other nodes that are part of the installation. Then Universal Installer runs `-attachHome` on the remote nodes to update the inventory. After this, you need to run `orainstRoot.sh` (if required) and `root.sh` on the local and remote nodes. After the scripts are run, the configuration steps are executed.

If the cluster is Cluster File System (CFS) or Network File System (NFS) mounted, Oracle Universal Installer does not propagate the Oracle home to other nodes, as files are shared across nodes and the installation will exist on the shared disk.

> **Note:** For cluster installations, you must run `oraInstRoot.sh` on each node of the cluster to set up the inventory.

Cluster Installation in Silent Mode

For Real Application Clusters installation, you can specify the `-local` flag, `CLUSTER_NODES`, `REMOTE_NODES`, and `LOCAL_NODE` session variables to indicate the nodes on which the installation needs to be done.

When you use the `-local` flag in a cluster installation, it means that the installation or inventory operation needs to be performed only on the local node. The `CLUSTER_NODES` session variable indicates that the specified nodes are part of the cluster.

```
./runInstaller -local "CLUSTER_NODES={a,b,c,d}"
```

In the above example, the installation would be performed in the local node, and nodes a, b, c, and d would be taken as the part of the cluster.

When you use the `REMOTE_NODES` session variable in a cluster installation, it means that the installation or the inventory operation needs to be performed on the local node and the other nodes apart from the local node.

```
./runInstaller "REMOTE_NODES={c,d}" "CLUSTER_NODES={a,b,c,d}"
```

In the above example, the installation would be performed on the local node a and nodes c and d. The nodes a, b, c, and d would be taken as the part of the cluster.

Cluster Deinstallation

When removing a clustered Oracle home, Oracle Universal Installer first removes software from the node from which you are deinstalling, then removes software from the other cluster nodes associated with the Oracle home.

Command Line Options for Cluster Installations (UNIX Only)

When using Oracle Universal Installer in suppressed or silent mode, specify the path to the remote copy or shell program on the local node to use for cluster installations using the following optional flags:

```
-remoteShell <path to program>
-remoteCopy <path to program>
```

The default for `remoteShell` is `/usr/local/bin/ssh`. The default for `remoteCopy` is `/usr/local/bin/scp`. If these are not present, Oracle Universal Installer defaults to `rsh` and `rcp`, respectively.

Adding Language Support for Installed Products

Oracle Universal Installer enables you to add language support for the products you have installed. To add a language definition for the products already installed in the Oracle Home, you use the `addLangs` flag of `runInstaller`.

User Interface-based Installation

1. From the shiphome or from the Oracle Home, start OUI by entering the following command and flag:

   ```
   ./runInstaller -addLangs
   ```

2. Select the product location of the shiphome.

3. Select the new language set from the languages panel. The language upgrade is performed for all the components in the shiphome that are located in the Oracle Home.

4. View the summary window with the new "Language Updates" node. The elements of this node provide the component information for which the language updates are performed.

Silent-based installation

In the installation phase, OUI copies the files for the languages to be installed.

1. Make sure the response file specifies the new languages. For silent installations, by default the language update is performed for the languages specified in the response file.

2. From the shiphome or from the Oracle Home, start OUI by entering the following command and flags:

   ```
   ./runInstaller -silent -addLangs
   ```

The language upgrade is performed for all the components in the shiphome that are located in the Oracle Home. In the installation phase, OUI copies the files for the languages to be installed.

Patchsets and Upgrades

Oracle Universal Installer enables you to upgrade a product from one version to another version. An upgrade is a major product enhancement that often requires installation of the upgraded software. For example, if you convert your Oracle *10g* Database to Oracle 11*g*R1 Database, then it is called an upgrade.

A group of patches form a patchset. For example, if you convert your Oracle 11*g*R1 from version 11.1.1 to version 11.1.2, then it is called applying a patchset.

Post-installation Tasks

After completing a Real Application Clusters installation, it is advisable that you perform the following tasks:

Verification

After your Oracle product with Real Application Clusters installation is complete, go to `$ORACLE_HOME/Opatch` on each node and execute the following command:

```
$ORACLE_HOME/OPatch/opatch lsinventory -detail
```

This command lists the components installed inside the Oracle home and also lists the local and remote nodes for the home. Verify this information. You may also want to look at installation logs `inventory.xml` and `oraclehomeproperties.xml`.

See in case the check fails.

Back Up Disk after Installation

After your Oracle product with Real Application Clusters installation is complete and after you are sure that your system is functioning properly, make a back-up of the contents of the disk.

Also make a back-up of the disk contents after you complete any node additions or node deletions and after running any de-installation procedures.

Download and Install Patches

Go to the OracleMetaLink Web site for required patches for your installation. To download the required patches:

1. Use a Web browser to view the OracleMetaLink Web site:

   ```
   http://metalink.oracle.com
   ```

2. Log in to OracleMetaLink.

3. On the main OracleMetaLink page, click **Patches**.

4. On the **Select a Patch Search Area** page, click **New MetaLink Patch Search**.

5. On the **Simple Search** page, click **Advanced**.

6. On the **Advanced Search** page, click the search icon next to the **Product or Product Family** field.

7. In the **Search and Select: Product Family** field, enter the Oracle product name in the **For** field and click **Go**.

8. Select the Oracle product under the Results heading and click **Select**. The product name appears in the **Product or Product Family** field, and the current release appears in the *Release* field.

9. Select your platform from the list in the **Platform** field and click **Go**.

10. Any available patches appear under the **Results** heading.

11. Click the number of the patch that you want to download.

12. On the **Patch Set** page, click **View README** and read the page that appears. The README page contains information about the patch set and how to apply the patches to your installation.

13. Return to the **Patch Set** page, click **Download**, and save the file on your system.

14. Use the unzip utility provided with Oracle software to uncompress the Oracle patches that you downloaded from OracleMetaLink. The unzip utility is located in the `$ORACLE_HOME/bin` directory.

Note: By default, the patches would only be applied on the nodes on which the Oracle software installation was done. You can change the node list by using the `-enableRollingUpgrade` option.

Configure Oracle Products

Many Oracle products and options must be configured before you use them for the first time. Before using individual Oracle products or options, refer to the manual in the product documentation library that is available on the documentation CD-ROM or on the Oracle Technology Network Web site.

Deinstalling Real Application Clusters Software

If you need to deinstall Real Application Clusters software, you must run the Oracle Universal Installer to deinstall the software on the same node from which you performed the installation. You must first deinstall the Oracle product before deinstalling the Oracle Clusterware software.

Perform the following procedures as described in the following sections to deinstall Oracle product and Oracle Clusterware software.

Deinstalling Oracle Product Software on UNIX

The following steps describe how to use Oracle Universal Installer to remove Oracle software from an Oracle home on UNIX systems:

Note: Always use Oracle Universal Installer to remove Oracle software. Do not delete any Oracle home directories without first using Oracle Universal Installer to remove the software.

---

1. If necessary, log in as the `oracle` user:

   ```
   $ su - oracle
   ```

2. Set the `ORACLE_HOME` environment variable to specify the path of the Oracle home directory that you want to remove:

   - Bourne, Bash, or Korn shell:

     ```
     $ ORACLE_HOME=/u01/app/oracle/product/11.2.0/db_1
     $ export ORACLE_HOME
     ```

   - C shell:

     ```
     $ setenv ORACLE_HOME /u01/app/oracle/product/11.2.0/db_1
     ```

3. Stop all processes running in this Oracle home.

4. Start Oracle Universal Installer using the Deinstall tool. For information on this tool, see "Removing Oracle Database Client Software" in the *Oracle Database Client Installation Guide 11g Release 2 (11.2) for Linux*.

Deinstalling Oracle Product Software on Windows

The following steps describe how to use Oracle Universal Installer to remove Oracle software from an Oracle home on Windows systems:

> **Note:** Always use Oracle Universal Installer to remove Oracle software. Do not delete any Oracle home directories without first using Oracle Universal Installer to remove the software.

1. Stop all the Oracle services running in this Oracle home.

2. Start Oracle Universal Installer: from the **Start** menu, select **Programs**, then **ORACLE_ HOME_NAME**, then **Oracle Installation Products**, then **Oracle Universal Installer**.

   The **Welcome** screen for Oracle Universal Installer appears.

3. Click **Deinstall Products**.

   The **Inventory** screen appears.

4. Expand the tree of installed components until you find the components to remove. For example, if you installed a database with the Enterprise Edition option and later installed additional components with the Custom option, expand the Oracle home component to display all the components installed in the Oracle home.

5. Select the components to remove.

6. Click **Remove**.

   The **Confirmation** screen appears.

7. Click **Yes** to remove the selected components to initiate the removal process.

8. After the components are removed from your computer, the **Inventory** screen appears without the removed components.

9. Click **Close** to close the **Inventory** screen.

10. Click **Cancel** to exit Oracle Universal Installer.

11. Click **Yes** to confirm that you want to exit.

    For detailed instructions on deinstalling an Oracle product, see the respective Oracle product installation guide.

Deinstalling Oracle Clusterware

Before deinstalling Oracle Cluster software, ensure that you have removed the respective Oracle product's home in each node by following the steps in the preceding section. The following section explains deinstallation on Windows and UNIX.

Deinstalling Oracle Clusterware from a UNIX Environment

Deinstall each Oracle product Real Application Clusters home by running the procedure in the previous section. Then complete the deinstallation by removing the Oracle Clusterware software using the following procedure:

1. Run the command `$CRS_HOME/install/rootdelete.sh` to disable the Oracle Clusterware applications running on the cluster node. The `rootdelete.sh` script requires three arguments. If you are running this command on a remote node of the cluster, use `remote` as the first argument; otherwise, use `local` as the first argument. If the `ocr.loc` file is on a shared file system, use `sharedvar`. Otherwise, use `nosharedvar` as the second argument. If the Oracle Clusterware home is on a shared file system, use `sharedhome`; otherwise, use `nosharedhome` as the third argument. Repeat this step on each node of the cluster from which you want to deinstall Oracle Clusterware.

2. Run the script `$CRS_HOME/install/rootdeinstall.sh` on a local node to remove the OCR. You only need to run this script once.

3. Start Oracle Universal Installer. In the **Welcome** page, click **Deinstall Products** to display the list of installed products. Select the Oracle Clusterware home to deinstall.

Deinstalling Oracle Clusterware from a Windows Environment

Deinstall each Oracle product Real Application Clusters home by running the procedure in the previous section. Then complete the deinstallation by removing the Oracle Clusterware software using one of the following procedures:

- Deinstalling Oracle Clusterware with No Previous Cluster Software Versions.

- Deinstalling Oracle Clusterware with Clusterware Downgrade to 10.2.

Deinstalling Oracle Clusterware with No Previous Cluster Software Versions Perform the following steps to deinstall Oracle 11*g* Clusterware software from a Windows environment:

1. Click **Start** and navigate to **Settings**, then to **Control Panel**, then to **Administrative Tools**, and then to **Services**. Stop the service `oracleremexec`.

2. Start the Oracle Universal Installer. On the **Welcome** page, click **Deinstall Products** to display the list of installed products. Select the Oracle Clusterware home you want to deinstall.

3. If you have services with names such as `OracleCRSTokenname`, remove them by running the following command:

```
crsuser remove user_name
where user_name is a user name.
```

4. Shut down and restart each node that is a member of your cluster.

5. If you are not using a cluster file system, on each node use Windows Explorer to delete the Oracle directory, its subdirectories, and their contents.

Deinstalling Oracle Clusterware with Clusterware Downgrade to 10.2 Perform the following steps to deinstall Oracle 11*g* Clusterware software from a Windows environment that also has 10.2 Real Application Clusters:

1. Run `$CRS_HOME\bin\GuiOracleOBJManager.exe` to make sure that the symbolic link named `srvcfg` exists and points to a disk partition, if you are not using OCFS to store the OCR.

2. If Oracle10*g* release 10.2 Clusterware uses Oracle CFS (Cluster File System), ensure the following is true for all nodes:

   - The registry key `HKLM\SOFTWARE\Oracle\osd9i\ocr` exists.

   - The registry key `HKLM\SOFTWARE\Oracle\osd9i\ocr` has a string value `CfsOcrRoot` that points to a release 10.2 OCR (CFS) location.

3. Run `$CRS_HOME\bin\ocrconfig -downgrade` to downgrade the Cluster Registry to a release 10.2 OCR.

4. On each node, copy `$CRS_HOME\cfs\OcfsFindVol.exe` to `%SYSTEMROOT%\system32\osd9i\cfs`.

5. Run `$CRS_HOME\oui\bin\setup.exe` to start Oracle Universal Installer. On the **Welcome** page, click **Deinstall Products** to list all the installed products. Select the Oracle Clusterware home name from the displayed products, and click **Remove** to deconfigure and deinstall the product.

6. On each node, run `%SYSTEMROOT%\system32\osd9i\olm\OracleOBJService.exe /install` to reinstall the Oracle 10.2 object service. Then start Oracle Object Service.

7. On each node, run `%SYSTEMROOT%\system32\osd9i\cfs\OcfsFindVol.exe /i:%SYSTEMROOT%\system32\osd9i\cfs\OcfsFindVol.exe` to reinstall the Oracle Cluster Volume service. Then start the Oracle Cluster Volume service.

8. From a 10.2 Real Application Clusters Oracle home on each node, run the command `ORACLE_HOME\bin\gsdservice.exe -install`. Then start the `OracleGSDService`.

9. On each node, copy `%SYSTEMROOT%\system32\osd9i\orafencedrv.sys` `%SYSTEMROOT%\system32\drivers\orafenceservice.sys`.

Converting Single-instance Nodes to Real Application Clusters

You can convert single-instance nodes to Real Application Clusters using Oracle Universal Installer. For details on converting single instance nodes to Real Application Clusters, see the *Oracle Real Application Clusters Installation Guide 11g Release 2 (11.2)* for your platform.

Troubleshooting Real Application Clusters / Oracle Clusterware Installation

Figure 5–1 explains how you can solve problems associated with the Real Application Clusters or Oracle Clusterware installation.

Figure 5–1   Troubleshooting Real Application Clusters / Oracle Clusterware Installation

6
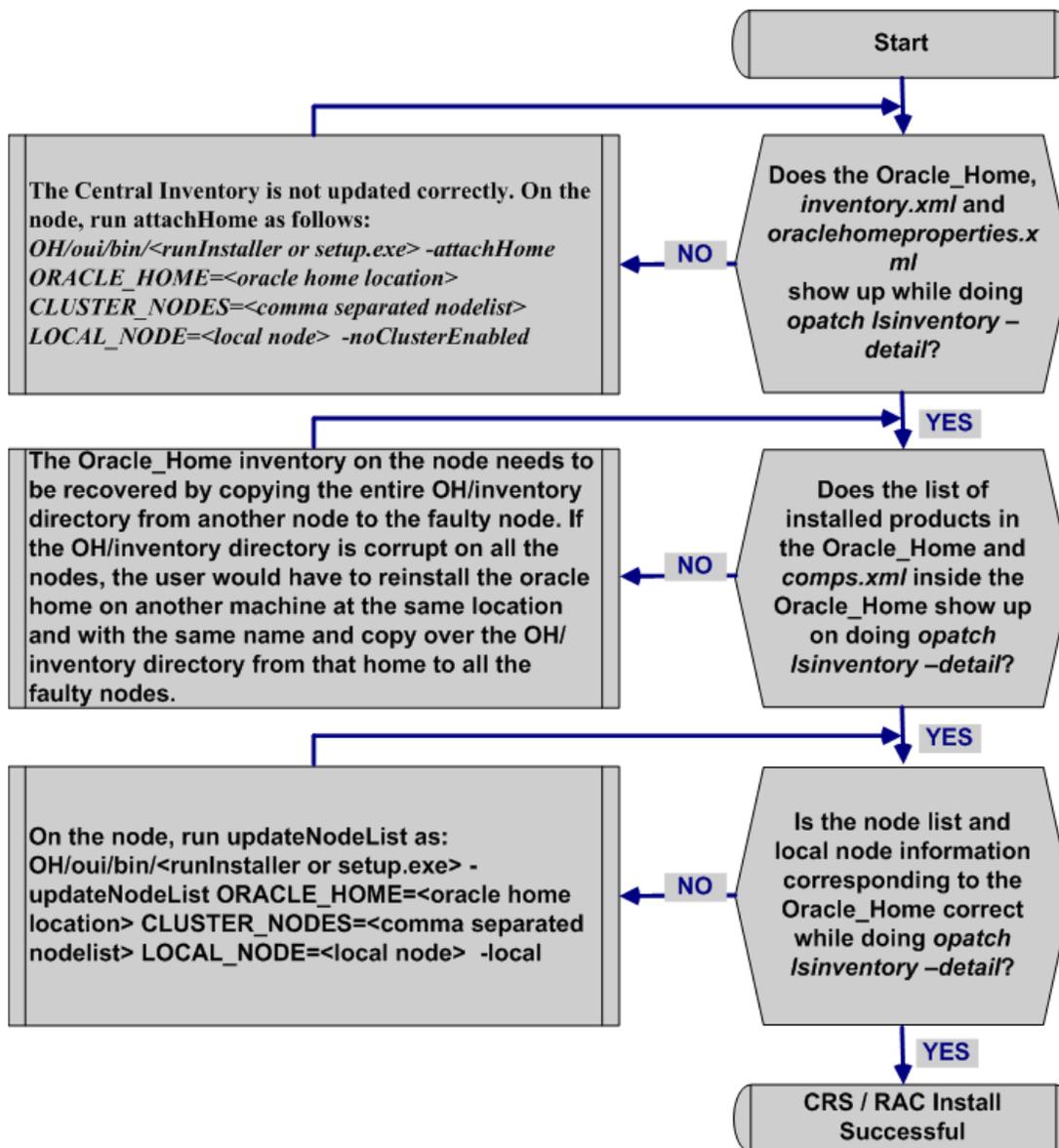
Cloning Oracle Software

This chapter provides information about Oracle cloning using Oracle Universal Installer (OUI). This chapter contains the following topics:

- About Cloning
- Overview of the Cloning Process
- Locating and Viewing Log Files

> **Note:** For procedures to clone an Oracle Database Oracle Home, refer to the following documentation depending on your platform:
>
> - Oracle® *Database Installation Guide for Linux* 11g Release 2 (11.2)
> - Oracle® *Database Installation Guide for Microsoft Windows* 11g Release 2 (11.2)
>
> For procedures to clone Oracle Clusterware and Oracle Real Application Clusters homes, refer to the following documentation:
>
> - *Oracle® Real Application Clusters Administration and Deployment Guide 11g Release 2 (11.2)*
> - *Oracle® Clusterware Administration and Deployment Guide 11g Release 2 (11.2)*

About Cloning

Cloning is the process of copying an existing Oracle installation to a different location and updating the copied bits to work in the new environment. The changes made by applying one-off patches on the source Oracle home are also present after the clone operation. The source and the destination path (host to be cloned) need not be the same. During cloning, OUI replays the actions that were run to install the home. Cloning is similar to installation, except that OUI runs the actions in a special mode referred to as clone mode. Some situations in which cloning is useful are:

- Creating an installation that is a copy of a production, test, or development installation. Cloning enables you to create a new installation with all patches applied to it in a single step. This contrasts with going through the installation process by performing separate steps to install, configure, and patch the installation.

- Rapidly deploying an instance and the applications that it hosts.

- Preparing an Oracle home and deploying it to many hosts.

The cloned installation behaves the same as the source installation. For example, the cloned Oracle home can be removed using OUI or patched using OPatch. You can also use the cloned Oracle home as the source for another cloning operation. You can create a cloned copy of a test, development, or production installation by using the command-line cloning scripts. The default cloning procedure is adequate for most usage cases. However, you can also customize various aspects of cloning, for example, to specify custom port assignments, or to preserve custom settings.

The cloning process copies all of the files from the source Oracle home to the destination Oracle home. Thus, any files used by the source instance located outside the source Oracle home's directory structure are not copied to the destination location.

The size of the binaries at the source and the destination may differ because these are relinked as part of the clone operation, and the operating system patch levels may also differ between these two locations. Additionally, the number of files in the cloned home would increase because several files copied from the source, specifically those being instantiated, are backed up as part of the clone operation.

OUI cloning is more beneficial than using the tarball approach, because cloning configures the Central Inventory and the Oracle home inventory in the cloned home. Cloning also makes the home manageable and enables the paths in the cloned home and the target home to be different.

Overview of the Cloning Process

The cloning process uses the OUI cloning functionality. This operation is driven by a set of scripts and add-ons that are included in the respective Oracle software. The cloning process has two phases:

- Source Preparation Phase
- Cloning Phase

Source Preparation Phase

To prepare the source Oracle home for cloning, archive and compress the source Oracle home using your preferred archiving tool. For example, you can use WinZip on Microsoft Windows system computers and `tar` or `gzip` on UNIX.

During this process, ensure the following:

- The tool that you use should preserve the permissions and file time stamps.

- When archiving the home, skip the `*.log`, `*.dbf`, `listener.ora`, `sqlnet.ora`, and `tnsnames.ora` files for archiving.

- Do not archive the following folders:

  ```
  $ORACLE_HOME/<Hostname>_<SID>
  $ORACLE_HOME/oc4j/j2ee/OC4J_DBConsole_<Hostname>_<SID>
  ```

Example 6–1 shows an example of an excluded file list.

Example 6–1   Excluded File List

```
$ cat excludedFileList.txt
./install/make.log
./cfgtoollogs/cfgfw/CfmLogger_2009-07-13_12-03-16-PM.log
./cfgtoollogs/cfgfw/oracle.server_2009-07-13_12-03-17-PM.log
./cfgtoollogs/cfgfw/oracle.network.client_2009-07-13_12-03-18-PM.log
./cfgtoollogs/cfgfw/oracle.has.common_2009-07-13_12-03-18-PM.log
```

```
./cfgtoollogs/cfgfw/oracle.assistants.server_2009-07-13_12-03-18-PM.log
./cfgtoollogs/cfgfw/OuiConfigVariables_2009-07-13_12-03-18-PM.log
./cfgtoollogs/cfgfw/oracle.sysman.console.db_2009-07-13_12-03-18-PM.log
./cfgtoollogs/cfgfw/oracle.sqlplus.isqlplus_2009-07-13_12-03-18-PM.log
./cfgtoollogs/oui/cloneActions2009-07-13_11-52-19AM.log
./cfgtoollogs/oui/silentInstall2009-07-13_11-52-19AM.log
```

Example 6–2 shows how to archive and compress the source for various platforms.

Example 6–2   Archiving and Compressing the Source

```
tar cpf - . | compress -fv > temp_dir/archiveName.tar.Z  ( for "aix" or $^O eq
"hpux")
tar cpfX - excludeListFile . | compress -fv > temp_dir/archiveName.tar.Z (for
remaining UNIX based systems)
```

> **Note:** Do not use the jar utility to archive and compress the Oracle home.

Cloning Phase

On the destination system, you unarchive the Oracle home and run the following OUI clone command from the destination home:

```
<destination home location>/oui/bin/runInstaller -clone -silent -defaultHomeName
ORACLE_HOME=<destination home location>
```

The following example shows how to unarchive and decompress the source for various platforms:

```
To unarchive:
mkdir  Destination_oracle_home
cd  Destination_oracle_home
zcat temp_dir/archiveName.tar.Z | tar xpf - (for "hpux")
zcat temp_dir/archiveName.tar.Z | tar xBpf -  (for remaining UNIX based systems)
```

> **Note:** The cloned home and source home will not be identical in size, because the cloned home will have additional files created during the cloning operation.

Locating and Viewing Log Files

The cloning script runs multiple tools, each of which may generate its own log files. However, the following log files that OUI and the cloning scripts generate are the key log files of interest for diagnostic purposes:

- **<Central_Inventory>/logs/cloneActions timestamp.log:** Contains a detailed log of the actions that occur during the OUI part of the cloning.

- **<Central_Inventory>/logs/oraInstall timestamp.err:** Contains information about errors that occur when OUI is running.

- **<Central_Inventory>/logs/oraInstall timestamp.out:** Contains other miscellaneous messages generated by OUI.

- **$ORACLE_HOME/clone/logs/clone timestamp.log:** Contains a detailed log of the actions that occur during the pre-cloning and cloning operations.

- **$ORACLE_HOME/clone/logs/error timestamp.log:** Contains information about errors that occur during the pre-cloning and cloning operations.

To find the location of the Oracle inventory directory:

On all UNIX system computers except Linux and IBM AIX, look in the `/var/opt/oracle/oraInst.loc` file. On IBM AIX and Linux-based systems, look in the `/etc/oraInst.loc` file.

On Windows system computers, you can obtain the location from the Windows Registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\INST_LOC`.

7

Patching Oracle Software with OPatch

OPatch is an Oracle-supplied utility that assists you with the process of applying interim patches to Oracle's software and rolling back interim patches from Oracle's software. This chapter provides information on using OPatch for these purposes.

This chapter includes the following topics:

- About OPatch
- Requirements for OPatch
- Prerequisite Checks for OPatch
- Backup and Recovery Considerations for Patching
- OPatch Utility for OUI-based Oracle Homes
- Standalone Patching
- Schema Patching
- Online Patching
- Oracle Real Application Clusters Patching
- About Patch Conflicts
- Problem Resolution

About OPatch

OPatch is a platform-dependent utility that requires installation of the Oracle Universal Installer.

Patches are a small collection of files copied over to an existing installation. They are associated with particular versions of Oracle products. When applied to the correct version of an installed product, patches result in an upgraded version of the product.

Interim patches are bug fixes available to customers in response to specific bugs. They require a particular base release or patchset to be installed before you can apply them. They generally address specific bugs for a particular customer. These patches are not versioned and are generally available in a future patchset as well as the next product release.

OPatch Features

The OPatch 11.2 utility has the following features:

- **Scalability** — OPatch is scalable to support a large number of patches.

- **Reliability —** OPatch is reliable and protects the Oracle home and inventory. It can bring back the Oracle home to a stable state from patch application failures. It can also easily detect patch conflicts.

- **Availability —** Opatch's online patching improves system availability by enabling database patches to be applied without needing to shut down databases.

- **Portability —** OPatch is compatible with all operating systems for which Oracle releases software.

- **Robust —** OPatch is very robust. It is very easy to apply a patch as well as remove it.

- **Easy to maintain —** OPatch is easy to maintain and is also extensible.

- **Support for Silent Operation —** OPatch supports silent operation. This mode enables you to run the software without any user interaction.

- **Support for Oracle Real Application Clusters —** OPatch supports Oracle Real Application Clusters and works well in that setup. It is easy to extend it to Enterprise Manager Grid Control.

- **Easy to debug —** OPatch has various levels of logging and tracing mechanisms. It also has a debug option that helps to easily diagnose software problems.

OPatch supports the following tasks:

- Applying an interim patch.

- Rolling back the application of an interim patch.

- Detecting a conflict when applying an interim patch after previous interim patches have been applied. It also suggests the best options to resolve a conflict.

- Reporting on installed products and interim patches.

Getting Interim Patches

Oracle releases interim patches frequently to fix a bug or a set of bugs. You can get the interim patches by doing the following:

1. Access and log into My Oracle Support at the following location:

   ```
   http://support.oracle.com
   ```

2. Click the **Patches & Updates** link.

3. Enter the **Patch or ID Number**, then click **Search**. A Patch Search Results table appears.

4. Using the Release and Platform columns, find the desired patch, then click the associated Patch ID.

5. In the page that now appears, click the **Download** button in the right-hand column.

Environment Variables OPatch Uses

OPatch uses the following environment variables:

ORACLE_HOME — Oracle home location.

PATH — Path information.

Requirements for OPatch

The OPatch utility requires the following environment:

- The Oracle home environment variable (`ORACLE_HOME`) must point to a valid Oracle home directory and match the value used during installation of the Oracle home directory.

- JRE version 1.4 or higher, Java commands for Windows, and `ar`, `cp`, `fuser`, and `make` commands for UNIX must be made available.

- The library path must be set correctly for Oracle Real Application Clusters environments. OPatch uses some APIs to detect if the system is an Oracle Real Application Clusters system. Ensure that the library path is set correctly as follows:

```
For Solaris:
LD_LIBRARY_PATH = $ORACLE_HOME/lib32:$ORACLE_HOME/lib

For HP-UX:
SHLIB_PATH=$ORACLE_HOME/lib32:/usr/lib
```

> **See Also:** For the latest information about the OPatch utility, to check for updates, and to get the latest versions, go to OracleMetaLink at:
>
> `http://www.oracle.com/support/metalink/index.html`

- You must have write permission to the files in the Oracle home before running the OPatch utility. Otherwise, OPatch fails.

Prerequisite Checks for OPatch

Before you invoke OPatch, perform the prerequisite checks described in the following sections.

Checks for Single Instances and Oracle Real Application Clusters

Check ORACLE_HOME and Environment Variable

OPatch verifies if the Oracle home is present. You must ensure that the `ORACLE_HOME` environment variable is set to the Oracle home of the product you are trying to patch. Check the respective vendor documentation for details to set the environment variable.

Check for JRE

OPatch requires JRE version 1.4 or higher to work properly.

Check for System Space

When OPatch processes the script for the installation of a patch, it simultaneously generates a Rollback script and saves a copy of every file edited or deleted during the patching. OPatch also backs up the inventory information. Consequently, Oracle recommends that you have sufficient system space to accommodate the patch and the backup information.

Check for Oracle Universal Installer and OPatch Version Compatibility

OPatch 11.2 requires Oracle Universal Installer 11.2 or higher to work properly. If the Oracle Universal Installer version is less than what OPatch requires, OPatch errors out.

Check for Patch Applicable on Operating System

OPatch detects if a particular patch is applicable for an operating system. If it is not applicable, OPatch displays an error message.

Check for System Commands

OPatch supports a set of properties used for various software operations. You can use these properties to control the internal operations of OPatch. By default, OPatch uses the standard Java property format to specify the properties. The following list shows the default properties and their values:

```
fuser=/sbin:/usr/sbin

ar=/usr/ccs/bin/

make=/usr/bin
```

You can specify OPatch properties in the following ways:

- By using the default OPatch properties.

- By specifying the location of the user-defined properties file.

- By using the command line. The syntax is as follows:

  ```
  PROPERTY_NAME=VALUE
  ```

  Example: `fuser=/sbin:/usr/sbin`

Additional Checks for Oracle Real Application Clusters

For Oracle Real Application Clusters, ensure that you perform the following prerequisite checks besides the other checks listed in the preceding section.

Check for User Equivalence

You must ensure that the cluster machines have user equivalence set for the user installing Oracle Clusterware/ Oracle Real Application Clusters. On UNIX, this means `rsh` or `ssh` or both should be set up on the cluster machines. On Windows, this means the same `<domain>\<user>` should have administrative privileges on all the cluster machines, and the machines should be a member of the `<domain>`.

If the user equivalence is set properly, the following command will work properly:

```
$ rsh <nodename> date
```

Check for OPatch Lsinventory

Ensure that you are able to invoke the `opatch lsinventory -detail` command and are able to see the node information being printed out. If you do not find the node information correctly printed out, you need to update the node list. For more information on updating the node list, see .

The following example shows the command output for the 134 installed products:

```
Invoking OPatch 11.1.0.6.6
```

```
Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation.  All rights reserved.


Oracle Home       : /scratch/usernameusername/11.2db
Central Inventory : /home/username/newDB/oraInventory
   from           : /etc/oraInst.loc
OPatch version    : 11.1.0.6.6
OUI version       : 11.2.0.1.0
OUI location      : /scratch/username/11.2db/oui
Log file location : /scratch/username/11.2db/cfgtoollogs/opatch/opatch2009-07-27_
10-18-36AM.log

Patch history file: /scratch/username/11.2db/cfgtoollogs/opatch/opatch_history.txt

Lsinventory Output file location :
/scratch/username/11.2db/cfgtoollogs/opatch/lsinv/lsinventory2009-07-27_
10-18-36AM.txt

--------------------------------------------------------------------------------
Installed Top-level Products (1):

Oracle Database 11g                                          11.2.0.1.0
There are 1 products installed in this Oracle Home.


Installed Products (134):

Agent Required Support Files                                 10.2.0.4.2
Assistant Common Files                                       11.2.0.1.0
Bali Share                                                   1.1.18.0.0
Buildtools Common Files                                      11.2.0.1.0
Character Set Migration Utility                              11.2.0.1.0
Cluster Verification Utility Common Files                    11.2.0.1.0
Database Configuration and Upgrade Assistants                11.2.0.1.0
Database SQL Scripts                                         11.2.0.1.0
Database Workspace Manager                                   11.2.0.1.0
Deinstallation Tool                                          11.2.0.1.0
Enterprise Edition Options                                   11.2.0.1.0
Enterprise Manager Agent                                     10.2.0.4.2
Enterprise Manager Agent Core Files                          10.2.0.4.2
Enterprise Manager Common Core Files                         10.2.0.4.2
Enterprise Manager Common Files                              10.2.0.4.2
Enterprise Manager Database Plugin -- Agent Support          11.2.0.1.0
Enterprise Manager Database Plugin -- Repository Support     11.2.0.1.0
Enterprise Manager Grid Control Core Files                   10.2.0.4.2
Enterprise Manager Minimal Integration                       11.2.0.1.0
Enterprise Manager plugin Common Files                       11.2.0.1.0
Enterprise Manager Repository Core Files                     10.2.0.4.2
Exadata Storage Server                                       11.2.0.1.0
Expat libraries                                               2.0.1.0.1
Generic Connectivity Common Files                            11.2.0.1.0
HAS Common Files                                             11.2.0.1.0
HAS Files for DB                                             11.2.0.1.0
Installation Common Files                                    11.2.0.1.0
Installation Plugin Files                                    11.2.0.1.0
Installer SDK Component                                      11.2.0.1.0
JAccelerator (COMPANION)                                     11.2.0.1.0
LDAP Required Support Files                                  11.2.0.1.0
OLAP SQL Scripts                                             11.2.0.1.0
Oracle 11g Warehouse Builder Required Files                  11.2.0.1.0
```

```
Oracle Advanced Security                             11.2.0.1.0
Oracle Application Express                           11.2.0.1.0
Oracle Call Interface (OCI)                          11.2.0.1.0
Oracle Clusterware RDBMS Files                       11.2.0.1.0
Oracle Code Editor                                   1.2.1.0.0I
Oracle Configuration Manager                         10.3.1.1.0
Oracle Configuration Manager Deconfiguration         10.3.1.0.0
Oracle Containers for Java                           11.2.0.1.0
Oracle Core Required Support Files                   11.2.0.1.0
Oracle Data Mining RDBMS Files                       11.2.0.1.0
Oracle Database 11g                                  11.2.0.1.0
Oracle Database 11g                                  11.2.0.1.0
Oracle Database 11g Multimedia Files                 11.2.0.1.0
Oracle Database Deconfiguration                      11.2.0.1.0
Oracle Database Gateway for ODBC                     11.2.0.1.0
Oracle Database User Interface                       2.2.13.0.0
Oracle Database Utilities                            11.2.0.1.0
Oracle Database Vault J2EE Application               11.2.0.1.0
Oracle Database Vault option                         11.2.0.1.0
Oracle DBCA Deconfiguration                          11.2.0.1.0
Oracle Display Fonts                                 9.0.2.0.0
Oracle Enterprise Manager Console DB                 11.2.0.1.0
Oracle Extended Windowing Toolkit                    3.4.47.0.0
Oracle Globalization Support                         11.2.0.1.0
Oracle Globalization Support                         11.2.0.1.0
Oracle Help For Java                                 4.2.9.0.0
Oracle Help for the  Web                             2.0.14.0.0
Oracle Ice Browser                                   5.2.3.6.0
Oracle Internet Directory Client                     11.2.0.1.0
Oracle Java Client                                   11.2.0.1.0
Oracle JDBC Server Support Package                   11.2.0.1.0
Oracle JDBC/OCI Instant Client                       11.2.0.1.0
Oracle JDBC/THIN Interfaces                          11.2.0.1.0
Oracle JFC Extended Windowing Toolkit                4.2.36.0.0
Oracle JVM                                           11.2.0.1.0
Oracle Label Security                                11.2.0.1.0
Oracle LDAP administration                           11.2.0.1.0
Oracle Locale Builder                                11.2.0.1.0
Oracle Message Gateway Common Files                  11.2.0.1.0
Oracle Multimedia                                    11.2.0.1.0
Oracle Multimedia Annotator                          11.2.0.1.0
Oracle Multimedia Client Option                      11.2.0.1.0
Oracle Multimedia Java Advanced Imaging              11.2.0.1.0
Oracle Multimedia Locator                            11.2.0.1.0
Oracle Multimedia Locator RDBMS Files                11.2.0.1.0
Oracle Net                                           11.2.0.1.0
Oracle Net Listener                                  11.2.0.1.0
Oracle Net Required Support Files                    11.2.0.1.0
Oracle Net Services                                  11.2.0.1.0
Oracle Netca Client                                  11.2.0.1.0
Oracle Notification Service                          11.2.0.0.0
Oracle Notification Service (eONS)                   11.2.0.1.0
Oracle ODBC Driver                                   11.2.0.1.0
Oracle ODBC Driverfor Instant Client                 11.2.0.1.0
Oracle OLAP                                          11.2.0.1.0
Oracle OLAP API                                      11.2.0.1.0
Oracle OLAP RDBMS Files                              11.2.0.1.0
Oracle One-Off Patch Installer                       11.2.0.0.2
Oracle Partitioning                                  11.2.0.1.0
Oracle Programmer                                    11.2.0.1.0
```

```
            Oracle Quality of Service Management (Client)                  11.2.0.1.0
            Oracle RAC Deconfiguration                                     11.2.0.1.0
            Oracle RAC Required Support Files-HAS                          11.2.0.1.0
            Oracle Real Application Testing                                11.2.0.1.0
            Oracle Recovery Manager                                        11.2.0.1.0
            Oracle Security Developer Tools                                11.2.0.1.0
            Oracle Spatial                                                 11.2.0.1.0
            Oracle SQL Developer                                           11.2.0.1.0
            Oracle Starter Database                                        11.2.0.1.0
            Oracle Text                                                    11.2.0.1.0
            Oracle Text Required Support Files                             11.2.0.1.0
            Oracle UIX                                                     2.2.24.5.0
            Oracle Universal Connection Pool                               11.2.0.1.0
            Oracle Universal Installer                                     11.2.0.1.0
            Oracle Wallet Manager                                          11.2.0.1.0
            Oracle XML Development Kit                                      11.2.0.1.0
            Oracle XML Query                                               11.2.0.1.0
            Parser Generator Required Support Files                        11.2.0.1.0
            Perl Interpreter                                               5.10.0.0.1
            Perl Modules                                                   5.10.0.0.1
            PL/SQL                                                         11.2.0.1.0
            PL/SQL Embedded Gateway                                        11.2.0.1.0
            Platform Required Support Files                                11.2.0.1.0
            Precompiler Common Files                                       11.2.0.1.0
            Precompiler Required Support Files                             11.2.0.1.0
            Provisioning Advisor Framework                                 10.2.0.4.2
            RDBMS Required Support Files                                   11.2.0.1.0
            RDBMS Required Support Files for Instant Client                11.2.0.1.0
            regexp                                                          2.1.9.0.0
            Required Support Files                                         11.2.0.1.0
            Sample Schema Data                                             11.2.0.1.0
            Secure Socket Layer                                            11.2.0.1.0
            SQL*Plus                                                       11.2.0.1.0
            SQL*Plus Files for Instant Client                             11.2.0.1.0
            SQL*Plus Required Support Files                                11.2.0.1.0
            SQLJ Runtime                                                   11.2.0.1.0
            SSL Required Support Files for InstantClient                   11.2.0.1.0
            Sun JDK                                                        1.5.0.17.0
            XDK Required Support Files                                     11.2.0.1.0
            XML Parser for Java                                            11.2.0.1.0
            XML Parser for Oracle JVM                                      11.2.0.1.0
            There are 134 products installed in this Oracle Home.

            There are no Interim patches installed in this Oracle Home.


            -------------------------------------------------------------------------------

            OPatch succeeded.
```

Backup and Recovery Considerations for Patching

> **Note:** It is highly recommended that you back up the `ORACLE_HOME`
> before any patch operation. You can back up the `ORACLE_HOME` using
> your preferred method. You can use any method such as `zip`,
> `cp -r`, `tar`, and `cpio` to compress the `ORACLE_HOME`.

If the `ORACLE_HOME` does not appear when you execute the `opatch lsinventory -detail` command, the `ORACLE_HOME` might be missing from the Central Inventory, or the Central Inventory itself could be missing or corrupted.

If the `ORACLE_HOME` is listed when you execute the `opatch lsinventory -detail` command, but the products and components within the `ORACLE_HOME` are not listed, the inventory within the `ORACLE_HOME` (local inventory) might be missing or corrupted.

If the local inventory is corrupted or lost for some reason, you can simply restore the `ORACLE_HOME/inventory` if it was backed up. If a backup does not exist, you may have to reinstall the software.

OPatch Utility for OUI-based Oracle Homes

You can run the OPatch utility, located in the `<Path_to_Oracle_Home>/OPatch` directory, with various commands and options. The following string shows the syntax for the OPatch utility:

```
<Path_to_OPatch>/opatch [-help] [-r[eport]] [command] [-option]
```

where:

- *help* — Displays the help message for the command.
- *report* — Prints the actions without executing.
- *command* — One of the OPatch commands, described in Table 7–1.
- *option* — One of the OPatch command options, described starting with Table 7–2.

Table 7–1   OPatch OUI-based Commands

Command	Description
apply	Installs an interim patch. See "Apply Command for OUI-based Oracle Homes" for more information.
napply	Applies Oracle Clusterware patches. See the "Auto Command for OUI-based Oracle Homes" for more information.
auto	Applies Oracle Clusterware patches. See the "Auto Command for OUI-based Oracle Homes" for more information.
lsinventory	Lists what is currently installed on the system. See "Lsinventory Command for OUI-based Oracle Homes" for more information.
query	Queries a given patch for specific details. See "Query Command for OUI-based Oracle Homes" for more information.
rollback	Removes an interim patch. See "Rollback Command for OUI-based Oracle Homes" for more information.
nrollback	Removes *n* number of patches (hence nrollback). See "Nrollback Command for OUI-based Oracle Homes" for more information.
version	Prints the current version of the patch tool. See "Version Command for OUI-based Oracle Homes" for more information.

To view additional information for any command, use the following command:

```
<Path_to_OPatch>/opatch command -help
```

If using Perl, use the following command:

```
perl opatch.pl command -help
```

To show the full syntax of the -help option, enter opatch -h to view the following display:

```
Usage: opatch [ -help ] [ -r[eport] ] [ command ]

            command := auto
                       apply
                       lsinventory
                       napply
                       nrollback
                       rollback
                       query
                       version
                       prereq
                       util

<global_arguments> := -help      Displays the help message for the command.
                      -report    Print the actions without executing.

example:
  'opatch -help'
  'opatch auto -help'
  'opatch apply -help'
  'opatch lsinventory -help'
  'opatch napply -help'
  'opatch nrollback -help'
  'opatch rollback -help'
  'opatch prereq -help'
  'opatch util -help'
```

Apply Command for OUI-based Oracle Homes

This command applies an interim patch to an Oracle home from the current directory. The ORACLE_HOME environment variable must be set to the Oracle home to be patched.

> **Note:** You must have write permission to the files in the Oracle home before using this command. Otherwise, OPatch fails.

Syntax

Use the following syntax for this command:

```
opatch apply  [-delay <value> ] [ -force ]
              [-invPtrLoc <Path to oraInst.loc> ]
              [-jre <LOC> ] [-local ] [-minimize_downtime ]
              [-no_bug_superset ] [-no_inventory ]
              [-oh <ORACLE_HOME> ] [-retry <value> ] [-silent ]
              [-verbose ]  [-no_relink] [-pre <parameters for the
              pre script in escaped double quotes> [-opatch_pre_end] ]
              [-post <parameters for the post script in escaped
              quotes> [-opatch_post_end] ] [-no_sysmod]
              [-property_file <Path to property file>]
              [-local_node <Local node name>]
              [-remote_nodes <List of remote nodes (node1,node2)>]
              [-connectString  <List of connect strings>]
              [-runSql] [-sqlScript <path of the sql file>]
              [-ptlSchema <portal schema>] [-ptlPassword <portal password>]
```

```
[-ptlConnect <portal connect string>]
[-init <parameters for the init script in escaped double
quotes> [-opatch_init_end] ] [-report]
[<Patch Location>]
```

Options

Table 7–2 lists the options available for this command.

Table 7–2   Apply Options for OUI Patches

Option	Description
connectString	Specifies the list of database instances on which the patch needs to be applied. Specify the value for this option using the following syntax:
	`SID:User:Passwd:Node`
	Example:
	`oracle:dba:dba:mymachine,oracle1:::`
	The SID is required, but you can disregard the other parameters if desired, because OPatch provides default values for them.
	Note: If the system is not part of an Oracle RAC setup and you want to patch just the local node, provide the node name as an empty string.
delay	Specifies how many seconds to wait before attempting to lock the inventory again for a previous failure. You can use this option only if you specify the `retry` option.
force	Removes conflicting patches from the system. If a conflict exists that prevents the patch from being applied, you can use this option to apply the patch. OPatch removes all the conflicting patches before applying the current patch.
init	Passes parameters to the `init` script, which executes before prerequisite checks are run. The values for this option must be enclosed in double-quotes.
invPtrLoc	Specifies the location of the `oraInst.loc` file. The `invPtrLoc` option is needed when this option is used during installation. Oracle recommends the use of the default Central Inventory for a platform.
jre	Instructs OPatch to use JRE (Java) from the specified location instead of the default location under the Oracle home directory.
local	Specifies that OPatch should patch the local node and update the inventory of the local node. It does not propagate the patch or inventory update to other nodes.
	You can use this option on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shut down before patching, you can use this option for non-rolling patches.
local_node	Tells OPatch the local node for this cluster. You can use this option on Oracle Real Application Clusters environments.
minimize_downtime	Specifies the order of nodes that OPatch should patch.
	This option only applies to Oracle Real Application Clusters environments. You cannot use it with the `-local` option with a rolling patch.
no_bug_superset	Specifies to error out if the current patch's bugs-to-fix is a superset (or same set) of an installed patch's bugs-fixed in the Oracle home directory.

Table 7–2   (Cont.) Apply Options for OUI Patches

Option	Description
no_inventory	Bypasses the inventory for reading and updates. You cannot use this option with the local option. This option places the installation into an unsupported state.
no_relink	This option does not perform any make operations. You can use it during multiple patch applications and to perform the linking step only once. OPatch does not keep track of the make operations it did not perform. You need to make sure to execute OPatch without this option at the end for compilation.
no_sysmod	Specifies that OPatch does not need to update the files in the system. It only updates the inventory. It also does not execute the pre and post scripts.
oh	Specifies the Oracle home directory to use instead of the default. This takes precedence over the environment variable ORACLE_HOME.
opatch_init_end	Marks the end of the init options. You use this option with the init option. If you do not use this option, everything after init until the end of the command is passed into init.
opatch_post_end	Marks the end of the post option. You use this option with the post option. If you do not use this option, everything after post until the end of the command is passed into post.
opatch_pre_end	Marks the end of the pre options. You use this option with the pre option. If you do not use this option, everything after pre until the end of the command is passed into pre.
Patch Location	Indicates the path to the patch location. If you do not specify the location, OPatch assumes the current directory is the patch location.
post	Specifies the parameters to be passed to the post script. This script is executed after the patch is applied. You need to enclose the values for this option in double-quotes.
pre	Specifies the parameters to be passed to the pre script. This script is executed before the patch is applied. You need to enclose the values for this option in double-quotes.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
ptlConnect	Specifies the connection string credentials of the portal schema.
ptlPassword	Specifies the password of the portal schema.
ptlSchema	Specifies the schema of the portal repository.
remote_nodes	Tells OPatch the list of remote nodes. You can use this option on Oracle Real Application Clusters environments. The node names must be separated with commas, but without spaces.
report	Prints the action to the screen without executing it.
retry	Tells OPatch how many times it should retry when there is an inventory lock failure.
runSql	Tells OPatch to run the SQL script and SQL procedures if they exist in the given patch. For information on SQL and PL/SQL patching, see "Schema Patching".
silent	Suppresses user interaction, and defaults any answers to "yes."

Table 7–2   (Cont.) Apply Options for OUI Patches

Option	Description
sqlScript	Specifies the custom SQL script that OPatch should run after patching completes. For information on SQL and PL/SQL patching, see "Schema Patching".
verbose	Prints additional OPatch output to the screen as well as to the log file.

> **Note:**   If a patch consists of SQL changes, follow the instructions in the patch readme, which is included with the patch to apply the SQL scripts.

Napply Command for OUI-based Oracle Homes

This command applies interim patches to several Oracle homes at the same time.

> **Note:**   You must have write permission to the files in the Oracle home before using this command. Otherwise, OPatch fails.

Syntax

Use the following syntax for this command:

```
opatch napply [patch_location] [-id comma-separated list of patch IDs]
                    [-delay <value> ] [ -force ]
                    [-invPtrLoc <Path to oraInst.loc> ]
                    [-jdk <LOC> ] [-jre <LOC> ] [ -local ]
                    [-minimize_downtime ] [-no_bug_superset ]
                    [-no_inventory ] [-oh <ORACLE_HOME> ]
                    [-retry <value> ] [-silent ]
                    [-verbose ]  [-no_relink]
                    [-pre <parameters for the pre script in
                    escaped double quotes> [-opatch_pre_end] ]
                    [-post <parameters for the post script in
                    escaped double quotes> [-opatch_post_end] ]
                    [-no_sysmod]
                    [ -property_file <Path to property file> ]
                    [ -local_node <Local node name> ]
                    [ -remote_nodes <List of remote nodes (node1,node2)> ]
                    [ -all_nodes ]
                    [ -phBaseFile <Path to the file containing the
                      location of the patches to be applied> ]
                    [-skip_subset] [-skip_duplicate] [-report]
```

Examples

- The following example applies all patches under the <patch_location> directory:

    ```
    opatch napply <patch_location>
    ```

- The following example applies patches 1, 2, and 3 that are under the <patch_location> directory:

    ```
    opatch napply <patch_location> -id 1,2,3
    ```

- The following example applies all patches under the `<patch_location>` directory. OPatch skips duplicate patches and subset patches (patches under `<patch_location>` that are subsets of patches installed in the Oracle home).

  ```
  opatch napply <patch_location> -skip_subset -skip_duplicate
  ```

  See the description for the `skip_subset` option in Table 7–3 for more information.

- The following example applies patches 1, 2, and 3 that are under the `<patch_location>` directory. OPatch skips duplicate patches and subset patches (patches under `<patch_location>` that are subsets of patches installed in the Oracle home).

  ```
  opatch napply <patch_location> -id 1,2,3 -skip_subset -skip_duplicate
  ```
  See the description for the `skip_subset` option in Table 7–3 for more information.

Options
Table 7–3 lists the options available for this command.

Table 7–3   Napply Options for OUI Patches

Option	Description
all_nodes	Applies the patch using the all-node mode.
delay	Specifies how many seconds to wait before attempting to lock the inventory again for a previous failure. You can use this option only if you specify the `retry` option.
force	Removes conflicting patches from the system. If a conflict exists that prevents the patch from being applied, you can use this option to apply the patch. OPatch removes all the conflicting patches before applying the current patch.
invPtrLoc	Specifies the location of the `oraInst.loc` file. The `invPtrLoc` option is needed when this option is used during installation. Oracle recommends the use of the default Central Inventory for a platform.
jdk	Instructs OPatch to use JDK (jar) from the specified location instead of the default location under the Oracle home directory. If you do not specify the `jre` option, JVM is executed from the `jdk` location.
jre	Instructs OPatch to use JRE (Java) from the specified location instead of the default location under the Oracle home directory. You cannot specify the `jdk` and `jre` options together.
local	Specifies that OPatch should patch the local node and update the inventory of the local node. It does not propagate the patch or inventory update to other nodes. You can use this option on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shut down before patching, you can use this option for non-rolling patches.
local_node	Tells OPatch the local node for this cluster. You can use this option on Oracle Real Application Clusters environments.
minimize_downtime	Specifies the order of nodes that OPatch should patch. This option only applies to Oracle Real Application Clusters environments. You cannot use it with the `-local` option with a rolling patch.

Table 7–3   (Cont.) Napply Options for OUI Patches

Option	Description
no_bug_superset	Specifies to error out if the current patch's bugs-to-fix is a superset (or same set) of an installed patch's bugs-fixed in the Oracle home directory.
no_inventory	Bypasses the inventory for reading and updates. You cannot use this option with the `local` option. This option places the installation into an unsupported state.
no_relink	This option does not perform any `make` operations. You can use it during multiple patch applications and to perform the linking step only once. OPatch does not keep track of the `make` operations it did not perform. You need to make sure to execute OPatch without this option at the end for compilation.
no_sysmod	Specifies that OPatch does not need to update the files in the system. It only updates the inventory. It also does not execute the pre and post scripts.
oh	Specifies the Oracle home directory to use instead of the default. This takes precedence over the environment variable `ORACLE_HOME`.
opatch_post_end	Marks the end of the `post` option. You use this option with the `post` option. If you do not use this option, everything after `post` until the end of the command is passed into `post`.
opatch_pre_end	Marks the end of the `pre` options. You use this option with the `pre` option. If you do not use this option, everything after `pre` until the end of the command is passed into `pre`.
Patch Location	Indicates the path to the patch location. If you do not specify the location, OPatch assumes the current directory is the patch location.
phBaseFile	If you do not specify `<patch_location>`, use this option to point OPatch to a file containing a list of patches to be n-applied. Each line in the file points to a location of a patch.
post	Specifies the parameters to be passed to the `post` script. This script is executed after the patch is applied. You need to enclose the values for this option in double-quotes.
pre	Specifies the parameters to be passed to the `pre` script. This script is executed before the patch is applied. You need to enclose the values for this option in double-quotes.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
remote_nodes	Tells OPatch the list of remote nodes. You can use this option on Oracle Real Application Clusters environments. The node names must be separated with commas, but without spaces.
report	Prints the action to the screen without executing it.
retry	Tells OPatch how many times it should retry when there is an inventory lock failure.
silent	Suppresses user interaction, and defaults any answers to "yes."
skip_duplicate	Skips patches to be applied that are duplicates of other patches installed in the Oracle home. Two patches are duplicates if they fix the same set of bugs.

Table 7–3 (Cont.) Napply Options for OUI Patches

Option	Description
skip_subset	Skips patches to be applied that are subsets of other patches installed in the Oracle home. One patch is a subset of another patch if the former fixes a subset of bugs fixed by the latter.
	For example, if you used `napply` yesterday for patch A that fixed bugs 1 and 2, then you use `napply` today with this option for patch B that fixes bug 1 and patch C that fixes bugs 1, 2, and 3, then subset patch A is skipped, and patch C then becomes a superset of patch A.
verbose	Prints additional OPatch output to the screen as well as to the log file.

Auto Command for OUI-based Oracle Homes

Ordinarily, an Oracle Clusterware patch requires several manual steps before and after you apply the patch, such as:

- Stopping all dependent databases

- Stopping Oracle Clusterware resources

- Running pre-patch scripts

- Shutting down Oracle Clusterware

- Running post-patch scripts

- Starting Oracle Clusterware and dependent databases

The opatch auto command automates all of these tasks for patching the grid infrastructure home and all other applicable RDBMS homes.

> **Note:** You must have write permission to the files in the Oracle home before using this command. Otherwise, OPatch fails.

Syntax

Use the following syntax for this command:

```
<path_to_Opatch>/opatch auto
        [-rollback [patch_location]]
        [[patch_location]-oh <path_to_oracle_home1>,<path_to_oracle_home2>...] |
        [[patch_location]-och <path_to_crs_home>]
```

... where *patch_location* is path to the location for the patch. If you do not specify the patch location, the current directory is considered the patch location.

Auto Options

Table 7–4 lists the options available for this command.

Table 7–4 Auto Options for OUI Patches

Option	Description
rollback	Rolls back the patch rather than applying it.
oh	Comma-separated Oracle homes to patch. The default is all applicable Oracle homes. Use this option to patch RDBMS homes where no database is registered.

Table 7–4   (Cont.) Auto Options for OUI Patches

Option	Description
och	Path of the Oracle Clusterware home. Use this option to patch only Oracle Clusterware homes where Oracle Clusterware has been stopped already. Do not use this option for Oracle Clusterware with a CRS stack that is up.

Examples

- The following example applies a patch with an unzipped patch location to all applicable Oracle homes on the system:

  ```
  opatch auto <patch_location>
  ```

- The following example rolls back the patch from all the applicable Oracle homes on the system:

  ```
  opatch auto -rollback <patch_location>
  ```

- The following example patches a selective list of Oracle homes:

  ```
  opatch auto <patch_location> -oh /tmp/oh1,/tmp/oh2,/tmp/oh3
  ```

- The following example only patches the CRS home when the Oracle Clusterware stack is down.

  ```
  opatch auto <patch_location> -och /tmp/ora_crs_home
  ```

Lsinventory Command for OUI-based Oracle Homes

This command lists the inventory for a particular Oracle home, or displays all installations that can be found. This command does not have any required options.

> **Note:**   You must have write permission to the files in the Oracle home before using this command. Otherwise, OPatch fails.

Syntax

Use the following syntax for this command:

```
 opatch lsinventory [-all ] [-all_nodes] [-bugs_fixed <asc | desc> ]
                    [-delay <value> ] [-detail ] [-group_by_date ]
                    [-inactive] [-invPtrLoc <Path to oraInst.loc> ]
                    [-jre <LOC> ] [-local ]
                    [-oh <ORACLE_HOME> ] [-patch <asc | desc> ]
                    [-ptlPassword <portal password> ]
                    [-ptlSchema <portal schema> ]
                    [-property_file <path to property file>]
                    [-retry <value> ]
                    [-retry <value> ] [-translation_patch ]
```

The following sections provide examples for the detail, bugs_fixed, and patch desc options. See Table 7–5 for descriptions of the command options.

-detail Option Example

The following example shows the output of `opatch lsinventory -detail` for 134 products and one interim patch:

```
Invoking OPatch 11.1.0.6.6
```

```
Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation.  All rights reserved.

Oracle Home       : /scratch/username/11.2db
Central Inventory : /home/username/newDB/oraInventory
   from           : /etc/oraInst.loc
OPatch version    : 11.1.0.6.6
OUI version       : 11.2.0.1.0
OUI location      : /scratch/username/11.2db/oui
Log file location : /scratch/username/11.2db/cfgtoollogs/opatch/opatch2009-07-27_
10-28-06AM.log

Patch history file: /scratch/username/11.2db/cfgtoollogs/opatch/opatch_history.txt

Lsinventory Output file location :
/scratch/username/11.2db/cfgtoollogs/opatch/lsinv/lsinventory2009-07-27_
10-28-06AM.txt

--------------------------------------------------------------------------------
Installed Top-level Products (1):

Oracle Database 11g                                          11.2.0.1.0
There are 1 products installed in this Oracle Home.


Installed Products (134):

Agent Required Support Files                                 10.2.0.4.2
Assistant Common Files                                       11.2.0.1.0
Bali Share                                                   1.1.18.0.0
Buildtools Common Files                                      11.2.0.1.0
Character Set Migration Utility                              11.2.0.1.0
Cluster Verification Utility Common Files                    11.2.0.1.0
Database Configuration and Upgrade Assistants                11.2.0.1.0
Database SQL Scripts                                         11.2.0.1.0
Database Workspace Manager                                   11.2.0.1.0
Deinstallation Tool                                          11.2.0.1.0
Enterprise Edition Options                                   11.2.0.1.0
Enterprise Manager Agent                                     10.2.0.4.2
Enterprise Manager Agent Core Files                          10.2.0.4.2
Enterprise Manager Common Core Files                         10.2.0.4.2
Enterprise Manager Common Files                              10.2.0.4.2
Enterprise Manager Database Plugin -- Agent Support          11.2.0.1.0
Enterprise Manager Database Plugin -- Repository Support     11.2.0.1.0
Enterprise Manager Grid Control Core Files                   10.2.0.4.2
Enterprise Manager Minimal Integration                       11.2.0.1.0
Enterprise Manager plugin Common Files                       11.2.0.1.0
Enterprise Manager Repository Core Files                     10.2.0.4.2
Exadata Storage Server                                       11.2.0.1.0
Expat libraries                                               2.0.1.0.1
Generic Connectivity Common Files                            11.2.0.1.0
HAS Common Files                                             11.2.0.1.0
HAS Files for DB                                             11.2.0.1.0
Installation Common Files                                    11.2.0.1.0
Installation Plugin Files                                    11.2.0.1.0
Installer SDK Component                                      11.2.0.1.0
JAccelerator (COMPANION)                                     11.2.0.1.0
LDAP Required Support Files                                  11.2.0.1.0
OLAP SQL Scripts                                             11.2.0.1.0
Oracle 11g Warehouse Builder Required Files                  11.2.0.1.0
```

```
            Oracle Advanced Security                          11.2.0.1.0
            Oracle Application Express                        11.2.0.1.0
            Oracle Call Interface (OCI)                       11.2.0.1.0
            Oracle Clusterware RDBMS Files                    11.2.0.1.0
            Oracle Code Editor                                1.2.1.0.0I
            Oracle Configuration Manager                      10.3.1.1.0
            Oracle Configuration Manager Deconfiguration      10.3.1.0.0
            Oracle Containers for Java                        11.2.0.1.0
            Oracle Core Required Support Files                11.2.0.1.0
            Oracle Data Mining RDBMS Files                    11.2.0.1.0
            Oracle Database 11g                               11.2.0.1.0
            Oracle Database 11g                               11.2.0.1.0
            Oracle Database 11g Multimedia Files              11.2.0.1.0
            Oracle Database Deconfiguration                   11.2.0.1.0
            Oracle Database Gateway for ODBC                  11.2.0.1.0
            Oracle Database User Interface                    2.2.13.0.0
            Oracle Database Utilities                         11.2.0.1.0
            Oracle Database Vault J2EE Application            11.2.0.1.0
            Oracle Database Vault option                      11.2.0.1.0
            Oracle DBCA Deconfiguration                       11.2.0.1.0
            Oracle Display Fonts                              9.0.2.0.0
            Oracle Enterprise Manager Console DB              11.2.0.1.0
            Oracle Extended Windowing Toolkit                 3.4.47.0.0
            Oracle Globalization Support                      11.2.0.1.0
            Oracle Globalization Support                      11.2.0.1.0
            Oracle Help For Java                              4.2.9.0.0
            Oracle Help for the  Web                          2.0.14.0.0
            Oracle Ice Browser                                5.2.3.6.0
            Oracle Internet Directory Client                  11.2.0.1.0
            Oracle Java Client                                11.2.0.1.0
            Oracle JDBC Server Support Package                11.2.0.1.0
            Oracle JDBC/OCI Instant Client                    11.2.0.1.0
            Oracle JDBC/THIN Interfaces                       11.2.0.1.0
            Oracle JFC Extended Windowing Toolkit             4.2.36.0.0
            Oracle JVM                                        11.2.0.1.0
            Oracle Label Security                             11.2.0.1.0
            Oracle LDAP administration                        11.2.0.1.0
            Oracle Locale Builder                             11.2.0.1.0
            Oracle Message Gateway Common Files               11.2.0.1.0
            Oracle Multimedia                                 11.2.0.1.0
            Oracle Multimedia Annotator                       11.2.0.1.0
            Oracle Multimedia Client Option                   11.2.0.1.0
            Oracle Multimedia Java Advanced Imaging           11.2.0.1.0
            Oracle Multimedia Locator                         11.2.0.1.0
            Oracle Multimedia Locator RDBMS Files             11.2.0.1.0
            Oracle Net                                        11.2.0.1.0
            Oracle Net Listener                               11.2.0.1.0
            Oracle Net Required Support Files                 11.2.0.1.0
            Oracle Net Services                               11.2.0.1.0
            Oracle Netca Client                               11.2.0.1.0
            Oracle Notification Service                       11.2.0.0.0
            Oracle Notification Service (eONS)                11.2.0.1.0
            Oracle ODBC Driver                                11.2.0.1.0
            Oracle ODBC Driverfor Instant Client              11.2.0.1.0
            Oracle OLAP                                       11.2.0.1.0
            Oracle OLAP API                                   11.2.0.1.0
            Oracle OLAP RDBMS Files                           11.2.0.1.0
            Oracle One-Off Patch Installer                    11.2.0.0.2
            Oracle Partitioning                               11.2.0.1.0
            Oracle Programmer                                 11.2.0.1.0
```

```
Oracle Quality of Service Management (Client)                11.2.0.1.0
Oracle RAC Deconfiguration                                   11.2.0.1.0
Oracle RAC Required Support Files-HAS                        11.2.0.1.0
Oracle Real Application Testing                              11.2.0.1.0
Oracle Recovery Manager                                      11.2.0.1.0
Oracle Security Developer Tools                              11.2.0.1.0
Oracle Spatial                                               11.2.0.1.0
Oracle SQL Developer                                         11.2.0.1.0
Oracle Starter Database                                      11.2.0.1.0
Oracle Text                                                  11.2.0.1.0
Oracle Text Required Support Files                           11.2.0.1.0
Oracle UIX                                                   2.2.24.5.0
Oracle Universal Connection Pool                             11.2.0.1.0
Oracle Universal Installer                                   11.2.0.1.0
Oracle Wallet Manager                                        11.2.0.1.0
Oracle XML Development Kit                                   11.2.0.1.0
Oracle XML Query                                             11.2.0.1.0
Parser Generator Required Support Files                      11.2.0.1.0
Perl Interpreter                                             5.10.0.0.1
Perl Modules                                                 5.10.0.0.1
PL/SQL                                                       11.2.0.1.0
PL/SQL Embedded Gateway                                      11.2.0.1.0
Platform Required Support Files                              11.2.0.1.0
Precompiler Common Files                                     11.2.0.1.0
Precompiler Required Support Files                           11.2.0.1.0
Provisioning Advisor Framework                               10.2.0.4.2
RDBMS Required Support Files                                 11.2.0.1.0
RDBMS Required Support Files for Instant Client              11.2.0.1.0
regexp                                                        2.1.9.0.0
Required Support Files                                       11.2.0.1.0
Sample Schema Data                                           11.2.0.1.0
Secure Socket Layer                                          11.2.0.1.0
SQL*Plus                                                     11.2.0.1.0
SQL*Plus Files for Instant Client                            11.2.0.1.0
SQL*Plus Required Support Files                              11.2.0.1.0
SQLJ Runtime                                                 11.2.0.1.0
SSL Required Support Files for InstantClient                 11.2.0.1.0
Sun JDK                                                       1.5.0.17.0
XDK Required Support Files                                   11.2.0.1.0
XML Parser for Java                                          11.2.0.1.0
XML Parser for Oracle JVM                                    11.2.0.1.0
There are 134 products installed in this Oracle Home.


Interim patches (1) :


Patch  103          : applied on Mon Jul 27 10:25:16 PDT 2009
   Created on 07 Nov 2007, 04:57:14 hrs US/Eastern
   Bugs fixed:
     103, 104, 105
   Files Touched:
     routeserver.ear --> ORACLE_HOME/md/jlib/routeserver.ear
   Patch Location in Inventory:
     /scratch/username/11.2db/inventory/oneoffs/103
   Patch Location in Storage area:
     /scratch/username/11.2db/.patch_storage/103_Nov_07_2007_04_57_14


-------------------------------------------------------------------------------


OPatch succeeded.
```

-bugs_fixed Option Example

The following example shows the output of `opatch lsinventory -bugs_fixed asc`:

```
Invoking OPatch 11.1.0.6.6

Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation.  All rights reserved.

Oracle Home       : /scratch/username/11.2db
Central Inventory : /home/username/newDB/oraInventory
   from           : /etc/oraInst.loc
OPatch version    : 11.1.0.6.6
OUI version       : 11.2.0.1.0
OUI location      : /scratch/username/11.2db/oui
Log file location : /scratch/username/11.2db/cfgtoollogs/opatch/opatch2009-07-27_
10-25-33AM.log

Patch history file: /scratch/username/11.2db/cfgtoollogs/opatch/opatch_history.txt

Lsinventory Output file location :
/scratch/username/11.2db/cfgtoollogs/opatch/lsinv/lsinventory2009-07-27_
10-25-33AM.txt
--------------------------------------------------------------------------------
Installed Top-level Products (1):

Oracle Database 11g                                          11.2.0.1.0
There are 1 products installed in this Oracle Home.

List of Bugs fixed by Installed Patches:

Bug       Fixed by  Installed at                 Description
          Patch
---       --------  ------------                 -----------

103       103       Mon Jul 27 10:25:16 PDT 2009  Try to Patch an ear
104       103       Mon Jul 27 10:25:16 PDT 2009  Dummy1
105       103       Mon Jul 27 10:25:16 PDT 2009  Dummy2

OPatch succeeded.
```

-patch desc Option Example

The following example shows the output of `opatch lsinventory -patch desc`:

```
Invoking OPatch 11.1.0.6.6

Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation.  All rights reserved.

Oracle Home       : /scratch/username/11.2db
Central Inventory : /home/username/newDB/oraInventory
   from           : /etc/oraInst.loc
OPatch version    : 11.1.0.6.6
OUI version       : 11.2.0.1.0
OUI location      : /scratch/username/11.2db/oui
Log file location : /scratch/username/11.2db/cfgtoollogs/opatch/opatch2009-07-27_
10-39-03AM.log

Patch history file: /scratch/username/11.2db/cfgtoollogs/opatch/opatch_history.txt
```

```
Lsinventory Output file location :
/scratch/username/11.2db/cfgtoollogs/opatch/lsinv/lsinventory2009-07-27_
10-39-03AM.txt

--------------------------------------------------------------------------------
Interim patches (2) :

Patch  101         : applied on Mon Jul 27 10:38:00 PDT 2009
   Created on 07 Nov 2007, 04:57:14 hrs US/Eastern
   Bugs fixed:
     101
   This patch overlays patches:
     103

Patch  103         : applied on Mon Jul 27 10:25:16 PDT 2009
   Created on 07 Nov 2007, 04:57:14 hrs US/Eastern
   Bugs fixed:
     103, 104, 105


--------------------------------------------------------------------------------

OPatch succeeded.
```

Lsinventory Options

Table 7–5 describes the options available for the lsinventory command.

Table 7–5   Lsinventory Options for OUI Patches

Option	Description
all	Reports the name and installation directory for each Oracle home directory found.
all_nodes	Reports the patches installed on the given Oracle home in all nodes of the Oracle RAC system. It also prints the Oracle binary's size and checksum on all nodes. You cannot use this option with the `all`, `detail`, or `patch` options.
bugs_fixed	Reports bugs fixed by installed patches in a tabular format. Besides the bugs fixed, the report also displays the installed patches, installed times, and bug descriptions.
	The fixed bugs are sorted per installed patch. Default display is patches in descending order based on installed time and ascending order of bugs within each patch. You can use 'asc' (or) 'desc' with this option to enforce sort order on bugs within each patch.
	You can use this option with the patch or patch option to obtain sort orders with installed patches.
delay	If you specify retry, this option tells OPatch how many seconds it should wait before attempting to lock the inventory again in case of a previous failure.
detail	Reports the installed products and other details. You cannot use this option with the `-all` option.
group_by_date	Specifies that OPatch should group all installed patches by the date they were installed in the Oracle home.
inactive	Lists all the inactive patches that have been superseded by other superset patch(es) installed in the Oracle home.

Table 7–5   (Cont.) Lsinventory Options for OUI Patches

Option	Description
invPtrLoc	Specifies the location of the `oraInst.loc` file. You need this option if you used the `invPtrLoc` option during the installation. Oracle recommends using the default Central Inventory for a platform.
jre	Specifies the location of a particular JRE (Java) to use instead of the default location under the Oracle home directory.
local	Instructs OPatch to only display inventory information for the local node.
oh	Specifies the Oracle home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.
patch	Lists the patch IDs installed in the Oracle home in ascending (asc) or descending (desc) order, which is the default, based on installed time.
ptlPassword	Enables you to specify the password of the portal schema.
ptlSchema	Enables you to specify the schema of the portal repository.
property_file	Indicates the user-defined property file that OPatch should use. The path to the property should be absolute. This property file takes precedence over the property file that OPatch supplies.
retry	Specifies how many times OPatch should retry when there is an inventory lock failure.
translation_patch	Displays the translation patch, language code, installed time of the translation patch, and base patch for any single row in the table.

Query Command for OUI-based Oracle Homes

This command queries a specific patch for specific details. It provides information about the patch and the system being patched.

> **Note:**   You must have write permission to the files in the Oracle home before using this command. Otherwise, OPatch fails.

Syntax

Use the following syntax for this command:

```
opatch query  [-all] [-is_auto_patch] [-is_translatable_patch]
                     [-get_base_bugs] [-get_component] [-get_os] [-get_date]
                     [-get_patch_language] [-get_patch_type] [-get_patch_model]
                     [-get_product_family] [-get_update_components] [-has_sql]
                     [-is_online_patch]  [-is_patchset_update]
                     [-is_portal_patch] [-is_rolling_patch]
```

Options

Table 7–6 lists the options available for the Query command.

Table 7–6   Query Options

Option	Description
all	Retrieves all information about a patch. This is equivalent to setting all available options.
get_base_bugs	Retrieves bugs fixed by the patch.
get_component	Retrieves components the patch affects.
get_date	Retrieves the patch creation date and time.
get_patch_language	Gets the language that the patch supports.
get_patch_model	Gets the patching model.
get_patch_type	Gets the patch type specified in the patch metadata.
get_product_family	Gets the product family specified in the patch metadata to which the patch belongs.
get_update_components	Gets all components to be upgraded in a patchset update.
has_sql	Indicates true if the patch has SQL-related actions. Otherwise, the option is false. For information on SQL and PL/SQL patching, see "Schema Patching".
is_auto_patch	Indicates true if the patch is auto-enabled. Otherwise, the option is false.
is_online_patch	Indicates true if the patch is an online patch. Otherwise, the option is false.
is_patchset_update	Indicates true if the patch is a patchset upate. Otherwise, the option is false.
is_portal_patch	Indicates true if the patch has portal actions. Otherwise, the option is false.
is_rolling_patch	Indicates true if the patch is a rolling patch. Otherwise, the option is false.
is_translatable_patch	Indicates true if the patch is translatable. Otherwise, the option is false.
oh	Specifies the Oracle home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.
Patch Location	Indicates the path to the patch location. If you do not specify the location, OPatch assumes the current directory is the patch location.

Rollback Command for OUI-based Oracle Homes

This command removes an existing one-off patch from the appropriate Oracle home directory indicated by the reference ID.

> **Note:**   You must have write permission to the files in the Oracle home before using this command. Otherwise, OPatch fails.

Syntax

Use the following syntax for this command:

```
opatch rollback -id <ID> [-ph <Patch Location>]  [-delay <value>]
               [-invPtrLoc <Path to oraInst.loc> ]
               [-jre <LOC> ] [-local]  [-oh <ORACLE_HOME>]
```

```
[-retry <value>]  [-silent]  [-verbose]
[-no_relink] [-pre <parameters for the pre
script in escaped double quotes> [-opatch_pre_end] ]
[-post <parameters for the post script in escaped
double quotes>[ -opatch_post_end] ] [-no_sysmod]
[-property_file <path to property file>]
[-local_node <Local node name>]
[-remote_nodes <List of remote nodes (node1,node2)>]
[-connectString  <List of connect strings>]
[-ptlSchema <portal schema>] [-ptlPassword <portal password>]
[-ptlConnect <portal connect string>]
[-runSql] [-sqlScript <path of the sql file>]
[-init <parameters for the init script in escaped double
quotes> [-opatch_init_end] ] [-report]
```

Options

Table 7–7 lists the options available for the Rollback command.

Table 7–7   Rollback Options for OUI Patches

Option	Description
all_nodes	Rolls back the patch using the all-nodes mode.
connectString	Specifies the list of database instances on which the patch needs to be applied. Specify the value for this option using the following syntax:
	`SID:User:Passwd:Node`
	Example:
	`oracle:dba:dba:mymachine,oracle1:::`
	The SID is required, but you can disregard the other parameters if desired, because OPatch provides default values for them.
	Note: If the system is not part of an Oracle RAC setup and you want to patch just the local node, provide the node name as an empty string.
delay	If you use the `retry` option with the `rollback` command, specifies how many seconds OPatch should wait before attempting to lock the inventory again if a previous failure occurs.
id	Indicates the patch to be rolled back. Use the `lsinventory` option to display all patch identifiers. Each one-off patch is indicated by its ID. To successfully roll back a patch, you must provide the patch identifier.
init	Passes parameters to the init script, which executes before prerequisite checks are run. The values for this option must be enclosed in double-quotes.
invPtrLoc	Specifies the location of the `oraInst.loc` file. You need to use this option if you used the `invPtrLoc` option during installation. Oracle recommends the use of the default Central Inventory for a platform.
jre	Specifies the location of a particular JRE (Java) for OPatch to use instead of the default location under the Oracle home directory.

Table 7–7   (Cont.) Rollback Options for OUI Patches

Option	Description
local	Specifies that OPatch roll back the local node, then update the inventory of the local node. It does not propagate the patch or inventory update to other nodes.
	You can use this option on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shut down before patching, you can use this option for non-rolling patches.
local_node	Specifies to OPatch that this is the local node for the cluster to be used for rollback.
	You can use this option for Oracle Real Application Clusters environments.
no_sysmod	Specifies that OPatch need not update the files in the system, only the inventory. It also does not execute the pre and post scripts.
no_relink	This option does not perform any make operation in the patch. You can use this option during multiple patch removals and to perform the compilation step only once.
oh	Specifies the Oracle home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.
opatch_init_end	Marks the end of the init options. Use this option with the init option. If you do not use this option, everything after init until the end of the command is passed into init.
opatch_post_end	Marks the end of the post options. Use this option with the post option. If you do not use this option, everything after post until the end of the command is passed into post.
opatch_pre_end	Marks the end of the pre options. Use this option with the pre option. If you do not use this option, everything after pre until the end of the command is passed into pre.
Patch Location	Indicates the path to the patch location. If you do not specify the location, OPatch assumes the current directory is the patch location.
ph	Specifies the valid patch directory area. Rollback uses the command types found in the patch directory to identify which commands are used for the current operating system.
post	Specifies the parameters to be passed inside the post script. This script executes after the patch is removed. You must enclose the value of this option in double-quotes.
pre	Specifies the parameters to be passed inside the pre script. This script executes before the patch is removed. You must enclose the value of this option in double-quotes.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
ptlConnect	Specifies the connection string credentials of the portal schema.
ptlSchema	Specifies the schema of the portal repository.
ptlPassword	Specifies the password of the portal schema.

Table 7–7   (Cont.) Rollback Options for OUI Patches

Option	Description
remote_nodes	Specifies to OPatch the list of remote nodes to be used for rollback of the patch. The node names must be separated with commas, but without spaces.
	You can use this option on Oracle Real Application Clusters environments.
report	Prints the actions to the screen without executing them.
retry	Instructs OPatch how many times it should retry when there is an inventory lock failure.
runSql	Instructs OPatch to run the SQL script and SQL procedures if they exist in the given patch. For information on SQL and PL/SQL patching, see "Schema Patching".
sqlScript	Specifies the custom SQL script that OPatch should run after patching completes. For information on SQL and PL/SQL patching, see "Schema Patching".
silent	Suppresses user interaction, and defaults any yes|no questions to "yes". An Oracle Real Application Clusters setup does not support this option.
verbose	Prints additional OPatch output to the screen as well as to the log file.

Nrollback Command for OUI-based Oracle Homes

This command rolls back interim patches from several Oracle homes at the same time.

> **Note:**   You must have write permission to the files in the Oracle home before using this command. Otherwise, OPatch fails.

Syntax

Use the following syntax for this command:

```
opatch nrollback -id <comma-separated list of patch IDs>
                    [-delay <value>] [-invPtrLoc <Path to oraInst.loc> ]
                    [-jdk <LOC> ] [-jre <LOC> ] [-local]
                    [-minimize_downtime] [-no_relink] [-oh <ORACLE_HOME> ]
                    [-retry <value>]  [-silent]  [-verbose]
                    [-pre <parameters for the pre script in
                    escaped double quotes> [-opatch_pre_end] ]
                    [-post <parameters for the post script in
                    escaped double quotes>[ -opatch_post_end] ]
                    [-no_sysmod] [-property_file <Path to property file>]
                    [-local_node <Local node name>]
                    [-remote_nodes <List of remote nodes (node1,node2)>]
                    [ -all_nodes ] [-report]
```

Example

The following example rolls back patches 1, 2, and 3 that have been installed in the Oracle home:

```
opatch nrollback -id 1,2,3
```

Options

Table 7–8 lists the options available for this command.

Table 7–8   Nrollback Options for OUI Patches

Option	Description
all_nodes	Rolls back the patch using the all-nodes mode.
delay	If you use the `retry` option with the `rollback` command, specifies how many seconds OPatch should wait before attempting to lock the inventory again if a previous failure occurs.
id	Indicates the patch to be rolled back. Use the `lsinventory` option to display all patch identifiers. Each one-off patch is indicated by its ID. To successfully roll back a patch, you must provide the patch identifier.
invPtrLoc	Specifies the location of the `oraInst.loc` file. You need to use this option if you used the `invPtrLoc` option during installation. Oracle recommends the use of the default Central Inventory for a platform.
jdk	Instructs OPatch to use JDK (jar) from the specified location instead of the default location under the Oracle home directory. If you do not specify the `jre` option, JVM is executed from the `jdk` location.
jre	Specifies the location of a particular JRE (Java) for OPatch to use instead of the default location under the Oracle home directory.
local	Specifies that OPatch roll back the local node, then update the inventory of the local node. It does not propagate the patch or inventory update to other nodes.
	You can use this option on Oracle Real Application Clusters environments and non-clustered environments. If an entire cluster is shut down before patching, you can use this option for non-rolling patches.
local_node	Specifies to OPatch that this is the local node for the cluster to be used for rollback.
	You can use this option for Oracle Real Application Clusters environments.
minimize_downtime	Specifies the order of nodes that OPatch should patch.
	This option only applies to Oracle Real Application Clusters environments. You cannot use it with the `-local` option with a rolling patch.
no_sysmod	Specifies that OPatch need not update the files in the system, only the inventory. It also does not execute the pre and post scripts.
no_relink	This option does not perform any `make` operation in the patch. You can use this option during multiple patch removals and to perform the compilation step only once.
oh	Specifies the Oracle home directory to use instead of the default directory. This takes precedence over the `ORACLE_HOME` environment variable.
opatch_post_end	Marks the end of the `post` options. Use this option with the `post` option. If you do not use this option, everything after `post` until the end of the command is passed into `post`.

Table 7–8   (Cont.) Nrollback Options for OUI Patches

Option	Description
opatch_pre_end	Marks the end of the pre options. Use this option with the pre option. If you do not use this option, everything after pre until the end of the command is passed into pre.
Patch Location	Indicates the path to the patch location. If you do not specify the location, OPatch assumes the current directory is the patch location.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
remote_nodes	Specifies to OPatch the list of remote nodes to be used for rollback of the patch. The node names must be separated with commas, but without spaces.
	You can use this option on Oracle Real Application Clusters environments.
report	Prints the actions to the screen without executing them.
retry	Instructs OPatch how many times it should retry when there is an inventory lock failure.
silent	Suppresses user interaction, and defaults any yes	no questions to "yes". A Real Application Clusters setup does not support this option.
verbose	Prints additional OPatch output to the screen as well as to the log file.

Version Command for OUI-based Oracle Homes

This command shows the current version number of the OPatch utility. Use the following syntax for this command:

```
<Path_to_OPatch>/opatch version
```

Standalone Patching

Standalone patching is available for Oracle homes that have not been installed using the Oracle Universal Installer. Standalone patching does not have Central Inventory registration, but still generates inventory files for the one-off inventory and future conflict checking. OPatch uses the presence of the OUI directory under ORACLE_HOME to determine whether it should operate in OUI-based or standalone mode.

The following sections discuss these standalone patching topics:

- Unsupported services for standalone patching
- Standalone patching requirements
- OPatch commands for standalone patching
- Use cases

Unsupported Services for Standalone Patching

Standalone patching provides most of the services that OUI-based patching provides. However, standalone patching does not provide the following services that OUI-based patching provides.

Looking up the component inventory

Standalone OPatch enables you to look up which patches have been applied to a standalone Oracle home, but it does not support looking up product components. For example, if you run `opatch lsinventory` on a JDeveloper Oracle Home, OPatch shows a list of patches applied on the home. It does not show which components the home has, however.

Looking up the Central Inventory

You cannot run `opatch lsinventory -all` to list all Oracle homes registered on the host (through the Central Inventory repository).

Migrating from standalone to OUI-based patching and vice versa

The assumption is that after you have installed a product as standalone without OUI, it remains standalone. For example, after having installed JDeveloper, you cannot put OUI (through copying or proper installation) onto the Oracle home and expect OPatch to treat the home as an OUI-based Oracle home.

Conversely, the assumption is that after you have installed a product with OUI, it remains OUI-based. For example, after you install Oracle RDBMS, you cannot remove OUI (either by removing or proper deinstallation) and expect OPatch to treat the home as a standalone Oracle home. OPatch will not work properly in this case and will corrupt the home.

Interoperating between standalone and OUI-based patches

Since you cannot migrate a home from standalone to OUI-based and vice versa, OPatch does not support interoperability between standalone and OUI-based Oracle homes.

Seamlessly working on a cloned standalone Oracle home

If you clone a standalone Oracle home S1 to another Oracle home OH2, Opatch will not function properly on the new cloned OH2.

Supporting RAC

OPatch relies on OUI to detect Oracle RAC and propagate files. Hence, standalone OPatch does not support RAC; it does not attempt to detect RAC, and its utility will not work. That is, OPatch always runs as `opatch apply -local`. OPatch does not support any patch propagation from one node to another node. Also, standalone OPatch does not support RAC-related utilities such as `opatch util runRemoteMake` (invokes relink on remote node).

Performing patch set operations

OPatch does not support patch set operations in either standalone or OUI modes. You need to use OUI for patch set operations.

Standalone Patching Requirements

Standalone patching requires the following environment:

- JRE version 1.4 or later

- Oracle home without OUI

- OPatch that supports standalone patching

All of the required files and directories must exist for OPatch to function correctly. If any of the files are missing, OPatch perceives that the patch has not been applied. You

would then have to take corrective action, returning the standalone inventory to a stable state.

OPatch Utility for Standalone Homes

As with OUI-based patching, you can run the OPatch utility, located in the `<Path_to_Oracle_Home>/OPatch` directory, with various commands and options. The following string shows the syntax for the OPatch utility:

```
<Path_to_OPatch>/opatch [-help] [-r[eport]] [command] [-option]
```

where:

- *help* — Displays the help message for the command.
- *report* — Prints the actions without executing.
- *command* — One of the OPatch commands.
- *option* — One of the OPatch command options.

Table 7–9 lists the commands available for standalone patching.

Table 7–9   OPatch Standalone Commands

Command	Description
apply	Installs an interim patch. See "Apply Command for Standalone OPatch" for more information.
lsinventory	Lists what is currently installed on the system. See "Lsinventory Command for Standalone OPatch" for more information.
query	Queries a given patch for specific details. See "Query Command for Standalone OPatch" for more information.
rollback	Removes an interim patch. See "Rollback Command for Standalone OPatch" for more information.
version	Prints the current version of the patch tool. See "Version Command for Standalone OPatch" for more information.

The following sections provide the syntax and options for each of these commands.

Apply Command for Standalone OPatch

The Apply command applies an interim patch to a standalone home from the current directory.

Syntax

Use the following syntax for this command:

```
opatch apply  [ -force ] [-jre <LOC> ] [-no_bug_superset ] [-no_inventory ]
              [-oh <ORACLE_HOME> ][-silent ][-verbose ]
              [-no_relink] [-pre <parameters for the
              pre script in escaped double quotes> [-opatch_pre_end] ]
              [-post <parameters for the post script in escaped
              quotes> [-opatch_post_end] ] [-no_sysmod]
              [-property_file <Path to property file>]
              [-init <parameters for the init script in escaped double
              quotes> [-opatch_init_end] ] [-report]
              [<Patch Location>]
```

Options

Table 7–10 lists the options available for the Apply command.

Table 7–10   Apply Options for Standalone Patches

Option	Description
force	Removes conflicting patches from the system by enabling you to change the product and version number of the standalone Oracle home. OPatch removes all the conflicting patches before applying the current patch.
init	Passes parameters to the init script, which executes before prerequisite checks are run. The values for this option must be enclosed in double-quotes.
jre	Instructs OPatch to use JRE (Java) from the specified location instead of the default location under the Oracle home directory.
no_bug_superset	Specifies to error out if the current patch's bugs-to-fix is a superset (or same set) of an installed patch's bugs-fixed in the Oracle home directory.
no_inventory	Bypasses the inventory for reading and updates. You cannot use this option with the local option. This option places the installation into an unsupported state.
no_relink	This option does not perform any make operations. You can use it during multiple patch applications and to perform the linking step only once. OPatch does not keep track of the make operations it did not perform. You need to make sure to execute OPatch without this option at the end for compilation.
no_sysmod	Specifies that OPatch does not need to update the files in the system. It only updates the inventory. It also does not execute the pre and post scripts.
oh	Specifies the Oracle home directory to use instead of the default. This takes precedence over the environment variable ORACLE_HOME.
opatch_init_end	Marks the end of the init options. You use this option with the init option. If you do not use this option, everything after init until the end of the command is passed into init.
opatch_post_end	Marks the end of the post option. You use this option with the post option. If you do not use this option, everything after post until the end of the command is passed into post.
opatch_pre_end	Marks the end of the pre options. You use this option with the pre option. If you do not use this option, everything after pre until the end of the command is passed into pre.
Patch Location	Indicates the path to the patch location. If you do not specify the location, OPatch assumes the current directory is the patch location.
post	Specifies the parameters to be passed to the post script. This script is executed after the patch is applied. You need to enclose the values for this option in double-quotes.
pre	Specifies the parameters to be passed to the pre script. This script is executed before the patch is applied. You need to enclose the values for this option in double-quotes.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
silent	Suppresses user interaction, and defaults any answers to "yes."

Table 7–10   (Cont.) Apply Options for Standalone Patches

Option	Description
verbose	Prints additional OPatch output to the screen as well as to the log file.

Lsinventory Command for Standalone OPatch

The Lsinventory command lists the inventory for a particular Oracle home, or displays all installations that can be found. This command does not have any required options.

Syntax

Use the following syntax for this command:

```
opatch lsinventory [-all ]  [-detail ] [-jre <LOC> ]
                [-oh <ORACLE_HOME> ] [-patch] [-oh]
                [-property_file <path to property file>]
```

Options

Table 7–12 lists the options available for the Lsinventory command.

Table 7–11   Lsinventory Options for Standalone Patches

Option	Description
all	Reports the name and installation directory for each Oracle home directory found.
detail	Reports the installed products and other details. You cannot use this option with the all option.
jre	Specifies the location of a particular JRE (Java) for OPatch to use instead of the default location under the Oracle home directory.
oh	Specifies the Oracle home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.
patch	Specifies the patches installed in the Oracle home.
property_file	Indicates the user-defined property file that OPatch should use. The path to the property should be absolute. This property file takes precedence over the property file that OPatch supplies.

Query Command for Standalone OPatch

This command queries a specific patch for specific details. It provides information about the patch and the system being patched.

Syntax

Use the following syntax for this command:

```
opatch query  [-all] [-jre <LOC> ] [-oh <LOC> ]
                [-get_component] [-get_os] [-get_date]
                [-get_base_bug] [-is_portal_patch]
                [-is_rolling_patch] [-is_online_patch]
                [-has_sql] [ <Patch Location> ]
```

Options

Table 7–12 lists the options available for the Query command.

Table 7–12   Query Options

Option	Description
all	Retrieves all information about a patch. This is equivalent to setting all available options.
get_base_bug	Retrieves bugs fixed by the patch.
get_component	Retrieves components the patch affects.
get_date	Retrieves the patch creation date and time.
has_sql	Indicates true if the patch has SQL-related actions. Otherwise, the option is false. For information on SQL and PL/SQL patching, see "Schema Patching".
is_online_patch	Indicates true if the patch is an online patch. Otherwise, the option is false.
is_portal_patch	Indicates true if the patch has portal actions. Otherwise, the option is false.
is_rolling_patch	Indicates true if the patch is a rolling patch. Otherwise, the option is false.
oh	Specifies the Oracle home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.
Patch Location	Indicates the path to the patch location. If you do not specify the location, OPatch assumes the current directory is the patch location.

Rollback Command for Standalone OPatch

The Rollback command removes an existing one-off patch from the appropriate Oracle home directory indicated by the reference ID.

Syntax

Use the following syntax for this command:

```
opatch rollback -id <ID> [-ph <Patch Location>]
             [-jre <LOC> ] [-oh <ORACLE_HOME>]
             [-silent]  [-verbose] [-no_relink]
             [-pre <parameters for the pre
             script in escaped double quotes> [-opatch_pre_end] ]
             [-post <parameters for the post script in escaped
             double quotes>[ -opatch_post_end] ] [-no_sysmod]
             [-property_file <path to property file>]
             [-init <parameters for the init script in escaped double
             quotes> [-opatch_init_end] ] [-report]
```

Options

Table 7–13 lists the options available for the Rollback command.

Table 7–13   Rollback Options for Standalone Patches

Option	Description
id	Indicates the patch to be rolled back. Use the lsinventory option to display all patch identifiers. Each one-off patch is indicated by its ID. To successfully roll back a patch, you must provide the patch identifier.

Table 7–13   (Cont.) Rollback Options for Standalone Patches

Option	Description
init	Passes parameters to the init script, which executes before prerequisite checks are run. The values for this option must be enclosed in double-quotes.
jre	Specifies the location of a particular JRE (Java) for OPatch to use instead of the default location under the Oracle home directory.
no_sysmod	Specifies that OPatch need not update the files in the system, only the inventory. It also does not execute the pre and post scripts.
no_relink	This option does not perform any make operation in the patch. You can use this option during multiple patch removals and to perform the compilation step only once.
oh	Specifies the Oracle home directory to use instead of the default directory. This takes precedence over the ORACLE_HOME environment variable.
opatch_init_end	Marks the end of the init options. Use this option with the init option. If you do not use this option, everything after init until the end of the command is passed into init.
opatch_post_end	Marks the end of the post options. Use this option with the post option. If you do not use this option, everything after post until the end of the command is passed into post.
opatch_pre_end	Marks the end of the pre options. Use this option with the pre option. If you do not use this option, everything after pre until the end of the command is passed into pre.
Patch Location	Indicates the path to the patch location. If you do not specify the location, OPatch assumes the current directory is the patch location.
ph	Specifies the valid patch directory area. Rollback uses the command types found in the patch directory to identify which commands are used for the current operating system.
post	Specifies the parameters to be passed inside the post script. This script executes after the patch is removed. You must enclose the value of this option in double-quotes.
pre	Specifies the parameters to be passed inside the pre script. This script executes before the patch is removed. You must enclose the value of this option in double-quotes.
property_file	Specifies the user-defined property file for OPatch to use. The path to the property file should be absolute. This property file takes precedence over the one that OPatch supplies.
report	Prints the actions to the screen without executing them.
silent	Suppresses user interaction, and defaults any yes|no questions to "yes". An Oracle Real Application Clusters setup does not support this option.
verbose	Prints additional OPatch output to the screen as well as to the log file.

Version Command for Standalone OPatch

This command shows the current version number of the OPatch utility. Use the following syntax for this command:

```
<Path_to_OPatch>/opatch version
```

Use Cases

The following sections provide scenarios that administrators can encounter when implementing standalone patching for the following types of operations:

- Inventory
- Patching
- Utility

Inventory Operations

The following tables explain the purpose of the use case along with preconditions and the process that occurs during the patching process.

Table 7–14   Getting Patch Information

Use Case Category	Description
Purpose	Show a list of interim patches installed on a standalone Oracle home.
Preconditions	$ORACLE_HOME is set and the Oracle home has been patched using the standalone OPatch.
User Input	Enter the following command: `opatch lsinventory`
OPatch Response	1. OPatch detects that this is a standalone Oracle home. 2. OPatch looks for the standalone inventory file. 3. OPatch prints out a list of installed interim patches.

Table 7–15   Getting Detailed Patch Information

Use Case Category	Description
Purpose	Show a detailed list of interim patches installed on a standalone Oracle home.
Preconditions	$ORACLE_HOME is set and the Oracle home has been patched using the standalone OPatch.
User Input	Enter the following command: `opatch lsinventory -detail`
OPatch Response	1. OPatch detects that this is a standalone Oracle home. 2. OPatch looks for the standalone inventory file. 3. OPatch prints out a list of installed interim patches as well as files affected by each interim patch.

Patching Operations

The following tables explain the purpose of the use case along with preconditions and the process that occurs during the patching process.

Table 7–16   Applying an Interim Patch - Case 1

Use Case Category	Description
Purpose	Apply an interim patch on a standalone Oracle home.
Preconditions	$ORACLE_HOME is set and the Oracle home has been patched using the standalone OPatch. The patch has been downloaded.

Table 7–16 (Cont.) Applying an Interim Patch - Case 1

Use Case Category	Description
User Input	Enter the following command:
	`opatch apply/patch_loc/123451`
OPatch Response	1. OPatch detects that this is a standalone Oracle home.
	2. OPatch looks for the standalone inventory file and checks for conflicts.
	3. OPatch performs an automatic rollback if there are conflicting patches.
	4. OPatch applies a new patch to the home.
	5. OPatch updates its standalone inventory.

Table 7–17 Applying an Interim Patch - Case 2

Use Case Category	Description
Purpose	Apply an interim patch on a standalone Oracle home that exists within another OUI-based Oracle home.
Preconditions	`$ORACLE_HOME` is set and the Oracle home has been patched using a new OPatch. The patch has been downloaded.
	The standalone Oracle home has a different directory patch than the OUI-based Oracle home. For example, the OUI-based Oracle home path is `/path`, whereas the standalone Oracle home is `/path/dev`.
User Input	Enter the following command:
	`opatch apply/patch_loc/123451`
OPatch Response	1. OPatch detects that this is a standalone Oracle home.
	2. OPatch looks for the standalone inventory file and checks for conflicts.
	3. OPatch performs an automatic rollback if there are conflicting patches.
	4. OPatch applies a new patch to the home.
	5. OPatch updates its standalone inventory.

Table 7–18 Applying an Interim Patch - Case 3

Use Case Category	Description
Purpose	Apply an interim patch on a standalone Oracle. It seems to be a standalone Oracle home, but OPatch detects it as OUI-based.
Preconditions	`$ORACLE_HOME` is set and the Oracle home has been patched using a new OPatch. The patch has been downloaded.
User Input	Enter the following command:
	`opatch apply/patch_loc/123451`
OPatch Response	1. OPatch detects `ORACLE_HOME/oui` and believes it is an OUI-based Oracle home.
	2. OPatch attempts to apply the patch as OUI-based Oracle home patching (if the patch is compatible with the home).

Table 7–19   Rolling Back an Applied Interim Patch

Use Case Category	Description
Purpose	Roll back an interim patch applied earlier on a standalone Oracle home.
Preconditions	`$ORACLE_HOME` is set and the Oracle home has been patched using a standalone OPatch.
User Input	Enter the following command: `opatch rollback -id 123451`
OPatch Response	1. OPatch detects this is a standalone Oracle home. 2. OPatch examines the standalone inventory file to determine if patch 123451 was applied. 3. OPatch rolls back patch 123451. 4. OPatch updates its standalone directory.

Utility Operations

The following tables explain the purpose of the use case along with preconditions and the process that occurs during the patching process.

Table 7–20   Loading an Arbitrary XML File

Use Case Category	Description
Purpose	Load an XML file, making sure it is XML-parsable.
Preconditions	`$ORACLE_HOME` is set.
User Input	Enter the following command: `opatch util loadXML` Note that the loadXML utility is often used as a debugging and troubleshooting tool.
OPatch Response	1. OPatch detects this is a standalone Oracle home. 2. OPatch prompts for the complete path to the XML file that you want to load. 3. OPatch opens the file and uses the XML parser to parse it. 4. OPatch reports that the file is XML-parsable.

Table 7–21   Verifying that the Patch is Applied

Use Case Category	Description
Purpose	Ensure that the patch was applied to the Oracle home.
Preconditions	`$ORACLE_HOME` is set, and the Oracle home has been patched using the standalone OPatch.
User Input	Enter the following command: `opatch util verify -ph/patch_loc/123451` Note that patch verification is automatically invoked when OPatch applies a patch to an Oracle home. You do not need to rerun `verify` after applying a patch.

Table 7–21 (Cont.) Verifying that the Patch is Applied

Use Case Category	Description
OPatch Response	1. OPatch detects this is a standalone Oracle home.
	2. OPatch examines `/patch_loc/123451` to make sure it is a valid patch area.
	3. OPatch examines the files in `/patch_loc/123451` to make sure the Oracle home was patched with the same bits.
	4. OPatch reports that both the patch inventory and patch binary are in the Oracle home.

Schema Patching

There are two types of schema patches:

- **SQL patch** — This patches the Oracle database with updated procedures and schema changes.

- **PL/SQL patch** — This also patches the Oracle database with updated procedures and schema changes, as for the SQL patch. However, a PL/SQL patch also mentions the procedure names in its patch metadata so that these procedures can be backed up for rollback.

The following sections discuss these topics:

- Schema patching options
- Standalone SQL execution

Schema Patching Options

Table 7–22 shows the schema patching options that OPatch supports for Apply and Rollback:

Table 7–22 Schema Patching Options

Option	Description
-runSql	Instructs OPatch to read the SQL script from the patch and run it on the specified SIDs. You must specify this option for the `patchmd.xml` SQL script specification and custom SQL script.
-sqlScript	Specifies OPatch to run this custom SQL script. This is an optional parameter.

-connectString	Provides a list of database instance SIDs, user, and password to be patched. Each entry is separated by a comma (,). The value for this option has the following format:
	
	`SID1:USER1:PASSWORD1:NODE1,`
	`SID2:USER2:PASSWORD2:NODE2`

Standalone SQL Execution

OPatch provides a utility to run only the SQL scripts to patch specified database instances. Use this utility only when you cannot apply or roll back SQL procedure actions using normal Apply or Rollback sessions.

The syntax for Apply is as follows:

```
opatch util applySql -id <patchIDs> -connectString <SID1:USER1:PASSWORD1:NODE1>
```

The syntax for Rollback is as follows:

```
opatch util rollbackSql –ph <patchLocation> (or) –phBaseFile <filename> (or)
–phBaseDir
        <dirname> -connectString <SID1:USER1:PASSWORD1:NODE1>
```

Online Patching

Regular patches typically contain `.o` (object) files and/or `.a` (archive) libraries, and therefore require a relink of the RDBMS binary. Online patches, however, contain `.so` files, which are dynamic/shared libraries, and do not require a relink of the RDBMS binary. Consequently, since a relink is not needed, you can apply or roll back online patches while the RDBMS instance is running. This simplifies administration, because no downtime is needed, and also results in a much quicker turnaround time for installing or de-installing Online Patches.

A regular RDBMS patch can require many minutes to install, since it requires instance shutdown, a relink, and instance startup. On the other hand, you can install an online patch in just a few seconds.

Online patches are only applicable for Oracle RDBMS and not any other products. Online patches are currently supported on the following Windows and UNIX platforms for version 11.2.0.1.0 and later:

- Linux x86

- Linux x86_64

- HP-UX Itanium

- Solaris SPARC 64-bit

- Solaris AMD 64-bit

- AIX (AIX 6.1 and later)

Oracle Real Application Clusters Patching

An Oracle Real Application Clusters environment enables active instances to concurrently execute transactions on a shared database. Patching in an Oracle Real Application Clusters environment is slightly different compared to patching a single node.

Interim Patching using OPatch follows a similar approach as that performed by Oracle Universal Installer to detect Oracle home and nodes of a cluster. OPatch interacts with the Oracle Universal Installer inventory through the Oracle Universal Installer Java SDK. If OPatch detects a cluster, it queries the inventory through Oracle Universal Installer to find the local node name and node list. If your node list is not updated, you can update it by using the `-updateNodeList` flag of Oracle Universal Installer. You can bypass remote actions using the `-local` flag, as shown below:

```
$ORACLE_HOME/oui/bin/<runInstaller or setup.exe> -updateNodeList ORACLE_
HOME=<oracle home location>
"CLUSTER_NODES={node1,node2,node3}"
-noClusterEnabled
-local
```

If you want to specify the local node or remote nodes of an Oracle Real Application Clusters setup to OPatch, you can use the `LOCAL_NODE` or `REMOTE_NODES` session variable and specify the node name(s), as shown below:

```
$ORACLE_HOME/oui/bin/<runInstaller or setup.exe> ORACLE_HOME=<oracle home
location>
```

```
"REMOTE_NODES={node1,node2,node3}"
LOCAL_NODE=<nodelist
for example:node1>
```

If OPatch does not automatically detect Oracle Real Application Clusters or its nodes, you need to investigate the contents of the inventory and ensure that it is complete.

You can patch Oracle Real Application Clusters in three different ways:

- All Node Patching
- Rolling Patching
- Minimum Downtime Patching

The following sections provide detailed information for these types of Oracle Real Application Clusters patching.

All Node Patching

Figure 7–1 shows a basic example of All Node Patching.

Figure 7–1   All Node Patching
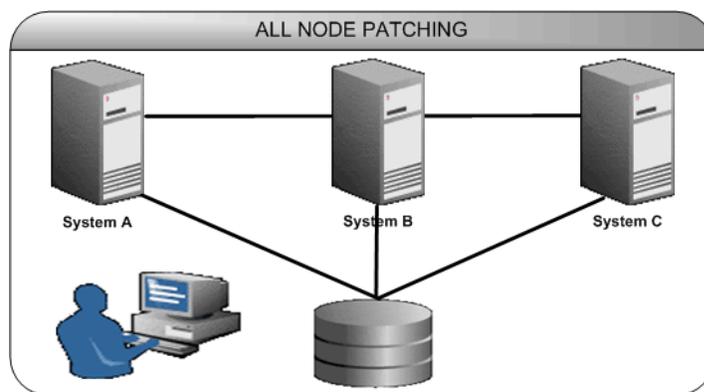
Systems A, B, and C are nodes in this cluster. When you perform All Node Patching in this cluster, you bring down systems A, B, and C, apply patches to all these nodes, then bring systems A, B, and C back up again.

Rolling Patching

In Rolling Patching, you shut down each node, apply the patch, then bring up each node again. You do this separately for each node until you patch all nodes in the cluster. This is the most efficient method of applying an interim patch to an Oracle Real Application Clusters setup, because there is absolutely no downtime during the application of patches, as only one system is brought down at any given time. Only some patches can be applied in this mode. The type is generally specified in the patch metadata.

Figure 7–2 shows a basic example of Rolling Patching.

Figure 7–2   Rolling Patching
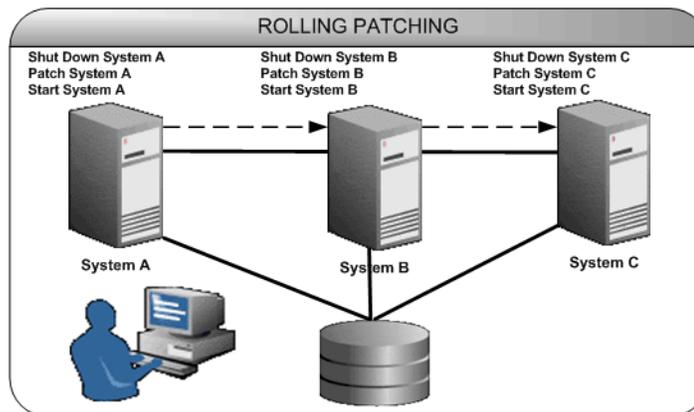
When you perform Rolling Patching in this cluster, the patches are applied in a rolling fashion. You initially bring down system A, apply a patch to it, then bring it back up. You do the same thing for systems B and C.

Minimum Downtime Patching

In Minimum Downtime Patching, the nodes are divided into sets. Initially, you shut down the first set and apply a patch to it. After this, you shut down the second set. You then bring up the first set and apply a patch to the second set. You now bring up the second set. All the nodes in the cluster are now patched. This method leads to less downtime for Oracle Real Application Clusters when both sets are brought down. This mode is executed by using -minimize_downtime command line option. You can also activate this option from the response file.

Figure 7–3 shows a basic example of Minimum Downtime Patching.

Figure 7–3   Minimum Downtime Patching
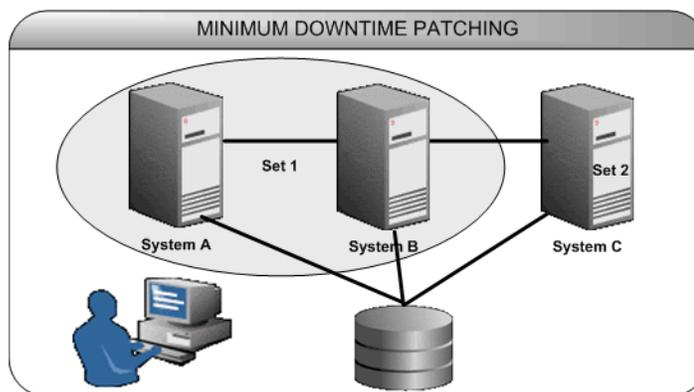
Systems A, B, and C are nodes in this cluster. It is divided into two sets: Set 1 contains systems A and B, and Set 2 contains system C. When you perform Minimum Downtime Patching in this cluster, you shut down Set 1 and apply a patch to it. You now shut down Set 2. Then, you bring up Set 1 and apply a patch to Set 2. After you apply the patch, you bring up Set 2 again. Now both Sets 1 and 2 are patched.

About Patch Conflicts

All patches may not be compatible with one another. For example, if you apply a patch, all the bugs the patch fixes could reappear after you apply another patch. This is called a conflict situation. OPatch detects such situations and raises an error when it detects a conflict.

Types of Conflicts

OPatch can detect the following types of conflicts.

Superset

If all the bugs fixed by a patch in the system are also fixed by the patch to be applied, this patch (the patch to be applied) is considered a superset of the patch already applied. If a bug superset condition is detected, it is not considered an error situation. All the subset patches are removed from the system and the new patch is applied.

Example

Consider the following scenario:

- Patch A, installed in the Oracle home, fixed bugs 1, 2, and 3.

- Patch B, installed in the Oracle home, fixed bugs 10, 11, and 12.

- Patch C, to be installed, fixes bugs 1, 2, 3, and 4.

Patch C is considered a superset of Patch A.

Using the -no_bug_superset Flag

If you want OPatch to error out if the current patch bugs-to-fix is a superset or the same as an installed patch bugs-fixed in the Oracle home directory, you can use the -no_bug_superset flag:

```
$ OPatch/opatch apply -no_bug_superset <Path_To_Patch>
```

The following example output shows the message you would see when you use the -no_bug_superset flag:

```
Oracle interim Patch Installer version 11.2.0.0.1
Copyright (c) 2009, Oracle Corporation.  All rights reserved.
Oracle Home       : /home/oracle_TEST/product/11.1.0/db_1
Central Inventory : /home/OUIHome_Opatch
from              : /home/oracle_TEST/product/11.2.0/db_1/oraInst.loc
OPatch version    : 11.1.0.6.6
OUI version       : 11.1.0.6.6
OUI location      : /home/oracle_TEST/product/11.1.0/db_1/oui
Log file location : /home/oracle_TEST/product/11.1.0/db
_1/cfgtoollogs/opatch/opatch-2008_May_25_14-03-33-IST_Wed.log
ApplySession applying interim patch '111000' to OH '/home/oracle
_TEST/product/11.1.0/db_1'
Apply Session failed: ApplySession failed to prepare the system. Interim patch
 111000 is a superset of the patch(es) [ 111000 ] in OH /home/oracle
_TEST/product/11.1.0/db_1
System intact, OPatch will not attempt to restore the system
OPatch failed with error code 73
```

Subset

Patches to be applied can be subsets of other patches installed in the Oracle home.

Example

Consider the following scenario:

- Patch A, installed in the Oracle home, fixed bugs 1, 2, and 3.
- Patch B, installed in the Oracle home, fixed bugs 10, 11, and 12.
- Patch D, to be installed, fixes bugs 1 and 2.

Patch D is a subset of Patch A.

Using the skip_subset Option

When you want to skip patches formerly applied in the Oracle home that are now subsets of other patches you want to apply now, you can use the `skip_subset` option of `napply`. For example, if you used `napply` yesterday for patch A that fixed bugs 1 and 2, then you use `napply` today with the `skip_subset` option for patch B that fixes bug 1 and patch C that fixes bugs 1, 2, and 3, then subset patch A is skipped, and patch C then becomes a superset of patch A.

Example 7–1 applies all patches under the `<patch_location>` directory. OPatch skips duplicate patches and subset patches (patches under `<patch_location>` that are subsets of patches installed in the Oracle home).

Example 7–1

```
opatch napply <patch_location> -skip_subset -skip_duplicate
```

Example 7–2 applies patches 1, 2, and 3 that are under the `<patch_location>` directory. OPatch skips duplicate patches and subset patches (patches under `<patch_location>` that are subsets of patches installed in the Oracle home).

Example 7–2

```
opatch napply <patch_location> -id 1,2,3 -skip_subset -skip_duplicate
```

See the description for the `skip_subset` option in Table 7–3 for more information.

Duplicate

A duplicate patch fixes the same set of bugs fixed by another patch. For example, if you applied Patch A that fixed bugs 1, 2 and 3, and now apply Patch B that also fixes bugs 1, 2 and 3, then Patch B is a duplicate of Patch A. A patch is always a duplicate of itself.

Using the skip_duplicate Option

If you specify this option, OPatch removes duplicate patches from the list of patches to be applied. For example, if you used `napply` yesterday for Patch A discussed above, then use `napply` today with the `-skip_duplicate` option for Patch A and other patches, duplicate Patch A is skipped.

Bug Conflict

A bug conflict occurs if a set of bugs to be fixed by the current interim patch intersects with some bugs already fixed by one or more previously installed interim patches. You must remove the bug conflict before you proceed with the patching by using the `apply` command with the `-force` flag, which rolls back the conflicting patches before applying the new one.

Example

Consider the following scenario:

- Patch A, installed in the Oracle home, fixed bugs 1, 2, and 3.

- Patch B, installed in the Oracle home, fixed bugs 10, 11, and 12.

- Patch E, to be installed, fixes bugs 3 and 4.

Patch E conflicts with Patch A.

File Conflict

A file conflict occurs if a set of files to be patched by the current interim patch includes files already patched by one or more previously installed interim patches, and it is not a bug superset.

Example

Consider the following scenario:

- Patch A, installed in the Oracle home, fixed bugs 1, 2, and 3, which modified files a, b, and c.

- Patch F, to be installed, fixes bugs 1, 2, 3 and 4, and modifies files a, d, and f.

Patch F conflicts with Patch A.

Patch Conflict Behavior for Apply and Napply

The expected behavior for the Apply and Napply commands is listed in Table 7–23.

Table 7–23 Expected Behavior for Apply and Napply Commands

Command	Superset	Subset	Duplicate	File Conflict or Bug Conflict Patch
Apply	OPatch performs an automatic rollback, then an apply.	After the merge request, OPatch performs an automatic rollback, then performs an apply.	OPatch performs an automatic rollback, then performs a reapply.	OPatch reports the conflict. After the merge request, OPatch performs an automatic rollback, then an apply.
Napply	OPatch performs an automatic rollback, then an apply.	OPatch reports the subset and skips the subset patch. It then continues and applies the other patches.	OPatch performs an automatic rollback, then a reapply.	OPatch reports the conflict, then asks you to run again without applying a bug conflict patch. You can use the -force option to instruct OPatch to automatically roll back the conflicting patch, then apply the new patch.

Patch Conflict Detection and Resolution

OPatch detects and reports any conflicts encountered when applying an Interim patch with a previously applied patch. The patch application fails in case of conflicts. You can use the -force option of OPatch to override this failure. If you use this option, the installer first rolls back any conflicting patches and then proceeds with the installation of the desired interim patch.

You may encounter a bug conflict and might want to remove the conflicting patch. This process is known as patch rollback. During patch installation, OPatch saves copies of all the files the new patch replaced before the new versions of these files are loaded and stores them in `$ORACLE_HOME/.patch_storage`. These saved files are called Rollback files and are the key to making patch rollback possible. When you roll back a patch, these Rollback files are restored to the system. You should only override the default behavior by using the `-force` flag if you completely understand the patch Rollback process. To roll back a patch, execute the following command:

```
$ OPatch/opatch rollback -id <Patch_ID>
```

Problem Resolution

The following sections provide information and instructions on the following tasks to resolve problems:

- Using logs and traces

- Recovering from a failed patching session

- Resolving OPatch application errors

Logging and Tracing

Logging and tracing is a common aid for debugging. OPatch maintains logs for all Apply, Rollback, and Lsinventory operations. Each time you execute OPatch, a new log file is created. The log files are located in the `<ORACLE_HOME>/cfgtoollogs/opatch` directory. Each log file is tagged with the timestamp of the operation. Log files are named as `opatch_<date mm-dd-yyyy>_<time hh-mm-ss>.log`.

For example, if a log file is created on May 17th, 2008 at 11.55 PM, it will be named as follows:

```
opatch_05-17-2008_23-55-00.log
```

> **Note:** You can set OPatch to debug mode by setting the environment variable `OPATCH_DEBUG` to `TRUE`.

Command Index

OPatch also maintains an index of the commands executed with OPatch and the log files associated with it in the `history.txt` file located in the `<ORACLE_HOME>/cfgtoollogs/opatch` directory. An example of the `history.txt` file is as follows:

```
Date & Time : Tue Apr 26 23:00:55 PDT 2009
Oracle Home : /private/oracle/product/11.2.0/db_1/
OPatch Ver. : 11.2.0.0.0
Current Dir : /scratch/oui/OPatch
Command     : lsinventory
Log File    :
/private/oracle/product/11.2.0/db_1/cfgtoollogs/opatch/opatch-2009_Apr_26_
23-00-55-PDT_Tue.log
```

Levels of Logging

OPatch follows the Oracle Diagnostic Logging (ODL) guidelines. You can set the log level by using the -logLevel <level> option available. This controls the amount of logging OPatch performs, according to the ODL guidelines.

OPatch supports the following log levels:

- SEVERE
- WARNING
- INFO
- CONFIG
- FINE
- FINER
- FINEST

Recovering from a Failed Patching Session

During patching, updates can occur in two phases:

- **System Update** — In this phase, the files are replaced in the Oracle home.
- **Inventory Update** — In this phase, the details of the patch applied is recorded in the inventory.

The following scenarios for single instance setups and Oracle Real Application Clusters setups explain how you can recover from a failed patching session.

Single Instance Setup

When you apply or roll back a patch, an interim inventory update exception occurs.

Cause: This occurs when the files on the system are patched, but the inventory update has failed. A corrupted inventory may cause this problem.

Action: Perform the following steps:

1. Ensure that the environment variable ORACLE_HOME is set properly.

2. Navigate to the $ORACLE_HOME/.patch_storage/<patch-id_timestamp> directory and execute the Restore command.

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
   ```

3. On UNIX, source $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt file (if available) as follows:

   ```
   /bin/sh make.txt
   ```

When you apply or roll back a patch, an OiiOneoffException occurs.

Cause: This occurs when the files on the system are patched, but the inventory update has failed. This may occur because the base component of the interim patch may not be present in the inventory.

Action: OPatch tries to restore the Oracle home automatically and displays a message for the same. If OPatch does not display a message stating that it has restored the Oracle home, perform the following steps:

1. Ensure that the environment variable `ORACLE_HOME` is set properly.

2. Navigate to the `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>` directory and execute the Restore command.

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
   ```

3. On UNIX, source `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt` file (if available) as follows:

   ```
   /bin/sh make.txt
   ```

When you apply a patch and execute opatch lsinventory, it returns nothing.

Cause: This may occur because all the patches applied before the application of the current patch are lost, or the patches might not have been updated in the inventory.

Action: Perform the following steps:

1. Navigate to the `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>` directory and execute the Restore command:

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
   ```

2. On UNIX, source `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt` file (if available) as follows:

   ```
   /bin/sh make.txt
   ```

3. If the files are properly patched, but the information is not updated in the inventory, execute the following command:

   ```
   $ORACLE_HOME/OPatch/opatch apply -no_sysmod <Path_To_Patch>
   ```

   Ensure that the patch has been applied and recorded properly in the inventory by executing the following command:

   ```
   $ORACLE_HOME/OPatch/opatch lsinventory -detail
   ```

4. If the files are still not patched properly, but you are able to see the patch in the `lsinventory` flag, you need to reapply the patch using the `no_inventory` flag:

   ```
   $ORACLE_HOME/OPatch/opatch apply -no_inventory <Path_To_Patch>
   ```

When you apply a patch and execute opatch lsinventory, it does not return the details of the patch applied.

Cause: OPatch may not have recorded the details of this patch in the inventory.

Action: Perform the following steps:

1. Navigate to the `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>` directory and execute the Restore command:

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
   ```

2. On UNIX, source `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt` file (if available) as follows:

   ```
   /bin/sh make.txt
   ```

3. If the files are properly patched, but the information is not updated in the inventory, execute the following command:

   ```
   $ORACLE_HOME/OPatch/opatch apply -no_sysmod <Path_To_Patch>
   ```

   Ensure that the patch has been applied and recorded properly in the inventory by executing the following command:

   ```
   $ORACLE_HOME/OPatch/opatch lsinventory -detail
   ```

When you press Ctrl + C during the application or roll back of a patch and execute opatch lsinventory, it does not return the details of the patch applied or rolled back.

Cause: This may be because OPatch might have stopped the application or rollback of the patch on pressing Ctrl+c.

Action: Perform the following steps:

1. Ensure that the environment variable `ORACLE_HOME` is set properly.

2. Navigate to the `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>` directory and execute the Restore command if it is available.

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
   ```

3. On UNIX, source `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt` file (if available) as follows:

   ```
   /bin/sh make.txt
   ```

When you apply a patch, you quit when OPatch failed to relink and prompted to continue.

Cause: This may occur because of a relink failure.

Action: Perform the following steps:

1. Ensure that the environment variable `ORACLE_HOME` is set properly.

2. Navigate to the `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>` directory and execute the Restore command.

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   ```

```
$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
```

3. Resolve the relink failure issue by ensuring that you are able to invoke `make` manually on a UNIX shell. After this, apply the patch again.

Oracle Real Application Clusters Setup

When I apply a patch on an Oracle Real Application Clusters setup and execute 'opatch lsinventory' on the local node, the patch is not listed.

Cause: This may occur if OPatch failed to update the inventory.

Action: Perform the following steps:

1. Ensure that the environment variable `ORACLE_HOME` is set properly in all the nodes of the cluster.

2. Navigate to the `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>` directory of each node in the cluster and execute the Restore command as follows:

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
   ```

3. On UNIX, source `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt` file (if available) in each node of the cluster as follows:

   ```
   /bin/sh make.txt
   ```

4. Apply the patch in each node in the cluster using the local flag:

   ```
   $ORACLE_HOME/OPatch/opatch apply -local <Path_To_Patch>
   ```

   > **Note:** Ensure that all the nodes use the same OPatch version.

When I apply a patch on an Oracle Real Application Clusters setup and execute 'opatch lsinventory' on the local node, it returns nothing.

Cause: You might have lost all the patches applied earlier.

Action: Perform the following steps:

1. Ensure that the environment variable `ORACLE_HOME` is set properly in each node in the cluster.

2. Navigate to the `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>` directory and execute the Restore command in each node in the cluster.

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
   ```

3. On UNIX, source `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt` file (if available) in each node as follows:

   ```
   /bin/sh make.txt
   ```

4. Apply the patch in each node using the local flag:

```
$ORACLE_HOME/OPatch/opatch apply -local <Path_To_Patch>
```

> **Note:** Ensure that all the nodes use the same OPatch version.

When I roll back a patch on an Oracle Real Application Clusters setup, and execute 'opatch lsinventory' on the local node, it shows that the patch was not removed.
Cause: This may occur if OPatch failed to update the inventory.

Action: Perform the following steps:

1. Ensure that the environment variable ORACLE_HOME is set properly in each node in the cluster.

2. Navigate to the $ORACLE_HOME/.patch_storage/<patch-id_timestamp> directory in each node in the cluster and execute the restore command as follows:

```
For UNIX:
$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

For Windows:
$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
```

3. On UNIX, source $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt file (if available) in each node in the cluster as follows:

```
/bin/sh make.txt
```

4. Roll back the patch in all the nodes in the cluster using the local flag:

```
$ORACLE_HOME/OPatch/opatch rollback -local -id <Patch_ID>
```

> **Note:** Ensure that all the nodes use the same OPatch version.

When I roll back a patch on an Oracle Real Application Clusters setup and execute 'opatch lsinventory' on the local node, it returns nothing.
Cause: You might have lost all the patches applied earlier.

Action: Perform the following steps:

1. Ensure that the environment variable ORACLE_HOME is set properly in each node in the cluster.

2. Navigate to the $ORACLE_HOME/.patch_storage/<patch-id_timestamp> directory and execute the Restore command in each node in the cluster:

```
For UNIX:
$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

For Windows:
$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
```

3. On UNIX, source $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt file (if available) as follows:

```
/bin/sh make.txt
```

4. Roll back the patch in the local node using the `local` flag:

    ```
    $ORACLE_HOME/OPatch/opatch rollback -local -id <Patch_ID>
    ```

5. Roll back the patch on the other nodes also using the `local` flag.

Note: Ensure that all the nodes use the same OPatch version.

When I apply a patch on an Oracle Real Application Clusters setup, the patching in one node is fine (both the files and the inventory are fine), but I am not sure about the other nodes.

Cause: This may occur because of a failed system or inventory update.

Action: Perform the following steps:

1. Copy the Oracle home from the node that is fine to the other nodes.

2. After copying the Oracle home, make sure that the `ORACLE_HOME/inventory/ContentsXML/comps.xml` file has the latest timestamp.

Note: On Unix, use `touch` to change the timestamp.

---

3. Update the nodes of the cluster. For more information on updating the nodes of the cluster, see "Updating the Nodes of a Cluster" on page 2-15.

4. Ensure that all the prerequisite checks pass that are listed in the section "Prerequisite Checks for OPatch" on page 7-3.

When I apply a patch on an Oracle Real Application Clusters setup, the patching in one node is fine, but when I execute 'opatch lsinventory' on the other nodes, the patch is not listed.

Cause: This may occur because of a failed system or inventory update.

Action: Perform the following steps:

1. Copy the `ORACLE_HOME /inventory` directory from the node that is fine to the other nodes.

2. After copying the `ORACLE_HOME /inventory` directory, make sure that the `ORACLE_HOME/inventory/ContentsXML/comps.xml` file has the latest timestamp.

Note: On Unix, use `touch` to change the timestamp.

---

3. Update the nodes of the cluster. For more information on updating the nodes of the cluster, see "Updating the Nodes of a Cluster" on page 2-15.

4. Ensure that all the prerequisite checks pass that are listed in the section "Prerequisite Checks for OPatch" on page 7-3.

When I apply or roll back a patch on an Oracle Real Application Clusters setup, I am not able to apply or roll back the patch on all nodes.

Cause: This may occur if the nodes are not properly updated.

Action: Perform any one or more of the following:

- Ensure that all the nodes in the cluster are up-to-date. If they are not, update the nodes of the cluster. For more information on updating the nodes of the cluster, see "Updating the Nodes of a Cluster" on page 2-15.

- Execute the appropriate command on all nodes of the cluster as follows:

```
opatch apply -local [patch_location]

opatch rollback -local [patch_location]
```

- Execute the appropriate command on the local node of the cluster as follows:

```
opatch apply [-local_node (node_name)] [-remote_nodes (comma separated
node_names)]

opatch rollback [-local_node (node_name)] [-remote_nodes (comma separated
node_names)]
```

Resolving OPatch Application Errors

This section provides solutions to the following errors that may occur during patch application:

- Not a valid patch area

- Opatch cannot find system commands like `fuser`, `make`

- Unable to remove a partially-installed interim patch

Not a valid patch area

Cause: The directory that the OPatch utility is using to do the patch does not match the template for what it is checking. This can also occur when you run the utility from an invalid shiphome directory.

Action: When starting the OPatch utility, the directory needs the following:

- `/etc` directory that has the metadata files.

- `/files` directory that has the payload files.

- `/etc/config/inventory` file and the actions file under the same directory.

If you did not start the OPatch utility from the `patch_id` directory, you can use the following command:

```
opatch apply /<Patch_Shiphome>
```

OPatch cannot find system commands like `fuser`, `make`

Cause: The OPatch utility uses `fuser` on UNIX systems to check for active Oracle instances. On certain hp-ux systems, only a super-user can run `fuser`.

Action: Perform these steps to resolve this problem:

1. Set `/tmp` in your PATH.

   For more information, see "Checks for Single Instances and Oracle Real Application Clusters" on page 7-3.

2. Create an empty file named `fuser`.

3. Shut down the Oracle instances.

4. Run the OPatch utility.

> **Caution:** Another way to resolve this problem is to give executable permission to other users for `fuser`. However, this exposes a potential security issue in the system, and is not recommended.

Unable to remove a partially-installed interim patch

Cause: Interruption in the patching process potentially causes this problem. This may occur if you press Ctrl+c during the patching process. If the error is the one that OPatch detects, it automatically resolves it.

Action: Perform the following steps:

1. Ensure that the environment variable `ORACLE_HOME` is set properly.

2. Navigate to the `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>` directory and execute the Restore command as follows:

   ```
   For UNIX:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.sh

   For Windows:
   $ORACLE_HOME/.patch_storage/<patch-id_timestamp>/restore.bat
   ```

3. On UNIX, source `$ORACLE_HOME/.patch_storage/<patch-id_timestamp>/make.txt` file (if available) as follows:

   ```
   /bin/sh make.txt
   ```

8

Oracle Internationalization and Translation

The following sections describe how Oracle Universal Installer supports multiple languages and internationalization:

- Installation Dialogs Language.
- Product Language Selections.
- Language Add-on

Installation Dialogs Language

Oracle Universal Installer runs in the language of the operating system. Oracle Universal Installer uses the language that Java detects, the system locale value, and sets that to the default language. The Oracle Universal Installer dialogs are displayed in this language if available. If specific Oracle Universal Installer dialogs are not translated in the language of the operating system, these dialogs are shown in English.

Oracle Universal Installer displays the translated GUI only if the variable `NLS_ENABLED` has been set to `TRUE` in the `oraparam.ini` file. If the `NLS_ENABLED` variable is set to `FALSE`, all text is shown in English.

Product Language Selections

If multiple languages are defined for the products installed, Oracle Universal Installer enables language selection for all installation types through the **Languages** button (shown on the Available Products page when there is more than one top-level component).

> **Note:** If there is only one top-level component, the **Languages** button is shown on the Install Types page.
>
> If the staging area has only one top-level component with only one installation type, the button is not shown and a Language Selection dialog is displayed later in the installation. You can also control the selected languages by setting the `SELECTED_LANGUAGES` variable in a response file or through the command line:
>
> ```
> setup.exe SELECTED_LANGUAGES={"fr","de"}
> ```

The language selections are for the top-level component being installed. A top-level component is the product selected on the Products Selection dialog, one per installation session. If any sub-components (not top-level) do not have files defined for

the language you select, only English only is installed. The English files are always installed for all components, regardless of user selection or operating system language.

During installation, Oracle Universal Installer examines the language list of each file and file group to determine which files need to be copied.

The Oracle Universal Installer product is translated into nine languages, as shown in Table 8–1, and includes translations for all generic Oracle Universal Installer dialogs. Note that any custom dialogs and product-specific information that are part of your installations must be translated and staged independently.

Table 8–1   *Oracle Universal Installer Translated Languages*

Language	Abbreviation
French	fr
Brazilian Portuguese	pt_BR
German	de
Italian	it
Japanese	ja
Korean	ko
Simplified Chinese	zh_CN
Spanish	es
Traditional Chinese	zh_TW

Language Add-on

Oracle Universal Installer 11.2 supports post-installation language translation additions. To get a version of an Oracle product with a particular language up and running, you need to invoke Oracle Universal Installer in `-addLangs` mode.

Perform the following steps to change the language of an installation using Oracle Universal Installer:

1. Execute the following command:

   ```
   ./runInstaller -addLangs
   ```

   A welcome page with a language add-on specific message appears.

2. Click **Next**. The Specify Home Details page appears.

3. Select the name and full path of the home containing the product to which you want to add languages and click **Next**. The Language Selection page appears.

4. Select the languages for which you want your Oracle products to be upgraded and click **Next**. A summary page displaying a snapshot of your preferences appears.

5. Review your preferences and click **Install**. After the languages are successfully updated, the End of Language Upgradation page appears.

6. Click **Exit**.

A

Frequently Asked Questions

This appendix answers the frequently asked questions for Oracle Universal Installer.

On Windows, when I run Oracle Universal Installer setup.exe, it is not invoked.

This may occur because some system resources are being held by some services. If you stop these services, Oracle Universal Installer can be invoked.

To run Oracle Universal Installer again, execute the following command:

```
setup -J-Dsun.java2d.noddraw=true -Dsun.awt.nopixfmt=true
```

When I start Oracle Universal Installer from any location, after setting the PATH environment variable, I receive an error that reads "Could not read any NLS message catalogue!". What do I do?

Oracle Universal Installer needs some files in the directory where the `runInstaller` (UNIX) or `setup.exe` (Windows) is running. So, when you invoke `runInstaller` (UNIX) or `setup.exe` (Windows), you should invoke it from the directory where this command is present, or you must specify the complete path.

I have lost my Central Inventory, but have a valid Oracle home(s). What do I do?

Oracle Universal Installer enables you to set up the Central Inventory or register an existing `ORACLE_HOME` with the Central Inventory in case of inventory corruption or loss. You need to execute the command with the `-attachHome` flag. For more information, see the section "Creating the Central Inventory" on page 2-8.

I want to deploy multiple Oracle home(s) from an existing Oracle home. How do I do this?

Oracle Universal Installer creates Oracle homes during Oracle software installation. To deploy multiple Oracle homes using an existing one, you need to clone the Oracle home. For more information on cloning and mass deployment, see Chapter 6, "Cloning Oracle Software".

I have lost my Oracle home inventory (comps.xml). What can I do?

Oracle recommends backing up the inventory when an Oracle home is installed or removed. Ensure that you back up the `comps.xml` that has the latest time-stamp. For more information on backing up the inventory, see "Backing up the Inventory" on page 2-13.

When your Oracle home inventory is corrupted or lost, you can restore from the backup or you can install the Oracle home from an identical installation.

When I run the configuration assistants, they fail. What do I do?

To recover from this problem, rename the configToolAllCommands and configToolFailedCommands files as follows:

- In UNIX, rename the configToolAllCommands and configToolFailedCommands files located in the OracleHome/cfgtoollogs/ directory to configToolAllCommands.sh and configToolFailedCommands.sh respectively and execute them.

- In Windows, rename the configToolAllCommands and configToolFailedCommands files located in the OracleHome/cfgtoollogs/ directory to configToolAllCommands.bat and configToolFailedCommands.bat respectively and execute them.

ConfigToolAllCommands(.sh/.bat) is used when the installation was done with a -noConfig option and you want to run all the configuration assistants in standalone mode, outside the OUI context.

ConfigToolFailedCommands(.sh/.bat) is used when the install was done with failed configuration assistants and you want to run only the failed configuration assistants later in standalone mode, outside the OUI context.

When I apply a patchset or an interim patch, the installer tries to propagate to a node that I disconnected a long time ago. What do I do?

This may occur because the Oracle home is present in the Central Inventory.

Oracle Universal Installer enables you to remove the Oracle home from the Central Inventory in cases where the Oracle home is uninstalled. To remove the Oracle home, you need to execute runInstaller with the -detachHome flag. You can also completely remove the Central Inventory to remove traces of log files. For more information, see the section "Detaching Oracle Homes from the Central Inventory" on page 2-9.

When I apply a patchset or an interim patch, the patch is not propagated to some of my Oracle Real Application Clusters nodes. What do I do?

In an Oracle Real Application Clusters environment, the inventory contains a list of nodes associated with an Oracle home. It is important that during the application of a patchset or an interim patch, the inventory is correctly populated with the list of nodes. If the inventory is not correctly populated with values, the patch is propagated only to some of the nodes in the cluster.

Oracle Universal Installer enables you to update inventory.xml with the nodes available in the cluster using the -updateNodeList flag in Oracle Universal Installer. For more information, see the section "Updating the Nodes of a Cluster" on page 2-15.

What if I just want to update the inventory without reapplying a one-off patch?

The OPatch addpatch.pl tool enables you to update the inventory with a given one-off patch without actually reapplying it to the Oracle home. Use this tool if you are certain that the patch is present in the Oracle home but not in the inventory. You can download it from MetaLink at patch number 4321390 to just add the patch to the inventory. After using this option, opatch lsinventory shows the one-off patch information along with other installed one-offs.

On two separate occasions, I experienced a corrupt inventory and a failed installation. Does OUI provide a backup facility I could use before I try to install again?

OUI provides automated backups for the Central Inventory and Oracle home Inventory. For more information, see Section , "Automated Inventory Backups" in Appendix B, "Troubleshooting and Debugging Oracle Universal Installer".

B

Troubleshooting and Debugging Oracle Universal Installer

This appendix is organized into the following troubleshooting sections:

- Debugging Mechanisms in Oracle Universal Installer
- Oracle Universal Installer Errors
- Other Tips

Debugging Mechanisms in Oracle Universal Installer

Oracle provides the following types of debugging mechanisms:

- Installation Log
- Automated Inventory Backups

The following sections discuss each of these mechanisms.

Installation Log

During the installation, Oracle Universal Installer writes a text file that contains information on:

- Variable settings
- Action calls
- Queries
- Exception information

The log of installation actions is written to a file named `installActions<date>.log`, located in the `<oraInventory>\logs` directory. The `oraInstall<timestamp>.err` and `oraInstall<timestamp>.out` files are also created and stored in the same logs directory. The `.log`, `.err`, and `.out` files for a particular session will be named with the same timestamp.

The installation log is used for debugging purposes only. All actions, queries, everything that happens during the installation, and all modifications to the target host are logged in the file.

Example B–1 shows a sample file:

Example B–1   Installation Actions File

```
Using paramFile: /scratch/oui/cd/Disk1/install/oraparam.ini

Checking swap space: must be greater than 500 MB.   Actual 4047 MB    Passed
Checking monitor: must be configured to display at least 256 colors.   Actual 256
Passed


The commandline for unzip:
/scratch/oui/cd/Disk1/install/unzip -qqqo
../stage/Components/oracle.swd.jre/1.5.0.11.0/1/DataFiles/\*.jar -d
//scratch/ohome/oracle/work/OraInstall2009-07-20_04-59-14AM
The commandline for unzip:
/scratch/oui/cd/Disk1/install/unzip -qqqo
../stage/Components/oracle.swd.oui/11.2.0.1.0/1/DataFiles/\*.jar -d
//scratch/ohome/oracle/work/OraInstall2009-07-20_04-59-14AM
The commandline for unzip:
/scratch/oui/cd/Disk1/install/unzip -qqqo
../stage/Components/oracle.swd.oui.core/11.2.0.1.0/1/DataFiles/\*.jar -d
//scratch/ohome/oracle/work/OraInstall2009-07-20_04-59-14AM
The commandline for unzip:
/scratch/oui/cd/Disk1/install/unzip -qqqo
../stage/Components/oracle.swd.opatch/11.2.0.0.2/1/DataFiles/\*.jar -d
//scratch/ohome/oracle/work/OraInstall2009-07-20_04-59-14AM


The number of files bootstrapped for the jre is 623.

The number of files bootstrapped for the oui is 77.
No umask is available from oraparam.ini. Using the default value '007'.


Execvp of the child jre : the cmdline is
//scratch/ohome/oracle/work/OraInstall2009-07-20_04-59-14AM/jre/1.5.0/bin/java,
and the argv is
//scratch/ohome/oracle/work/OraInstall2009-07-20_04-59-14AM/jre/1.5.0/bin/java
-Doracle.installer.library_loc=//scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/lib/linux
-Doracle.installer.oui_loc=//scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui
-Doracle.installer.bootstrap=TRUE
-Doracle.installer.startup_location=/scratch/oui/cd/Disk1/install
-Doracle.installer.jre_loc=//scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/jre/1.5.0
-Doracle.installer.nlsEnabled="TRUE"
-Doracle.installer.prereqConfigLoc=
-Doracle.installer.unixVersion=2.6.9-78.0.0.0.1.ELsmp
-mx96m
-cp
//scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM::/scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/OraInstaller.jar://scratch/ohome/oracle/work/OraInstall2009-07
-20_
04-59-14AM/oui/jlib/oneclick.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/xmlparserv2.jar://scratch/ohome/oracle/work/OraInstall2009-07-
20_04-59-14AM/oui/jlib/share.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/OraInstallerNet.jar://scratch/ohome/oracle/work/OraInstall2009
-07-20_
04-59-14AM/oui/jlib/emCfg.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/emocmutl.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/OraPrereq.jar://scratch/ohome/oracle/work/OraInstall2009-07-20
_04-59-14AM/oui/jlib/jsch.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/ssh.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
```

```
04-59-14AM/oui/jlib/remoteinterfaces.jar://scratch/ohome/oracle/work/OraInstall200
9-07-20_04-59-14AM/oui/jlib/http_
client.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/OPatch/jlib/opatch.jar://scratch/ohome/oracle/work/OraInstall2009-07-20
_
04-59-14AM/OPatch/jlib/opatchactions.jar://scratch/ohome/oracle/work/OraInstall200
9-07-20_
04-59-14AM/OPatch/jlib/opatchprereq.jar://scratch/ohome/oracle/work/OraInstall2009
-07-20_
04-59-14AM/OPatch/jlib/opatchutil.jar://scratch/ohome/oracle/work/OraInstall2009-0
7-20_
04-59-14AM/oui/jlib/OraCheckPoint.jar://scratch/ohome/oracle/work/OraInstall2009-0
7-20_
04-59-14AM/oui/jlib/InstImages.jar://scratch/ohome/oracle/work/OraInstall2009-07-2
0_
04-59-14AM/oui/jlib/InstHelp.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_
de.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_
es.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_
fr.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_
it.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_
ja.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_
ko.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_pt_
BR.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_zh_
CN.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/InstHelp_zh_
TW.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/oracle_
ice.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/help4.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/help4-nls.jar://scratch/ohome/oracle/work/OraInstall2009-07-20
_04-59-14AM/oui/jlib/ewt3.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/ewt3-swingaccess.jar://scratch/ohome/oracle/work/OraInstall200
9-07-20_
04-59-14AM/oui/jlib/ewt3-nls.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/swingaccess.jar:://scratch/ohome/oracle/work/OraInstall2009-07
-20_
04-59-14AM/oui/jlib/jewt4.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/jewt4-nls.jar://scratch/ohome/oracle/work/OraInstall2009-07-20
_
04-59-14AM/oui/jlib/orai18n-collation.jar://scratch/ohome/oracle/work/OraInstall20
09-07-20_
04-59-14AM/oui/jlib/orai18n-mapping.jar://scratch/ohome/oracle/work/OraInstall2009
-07-20_
04-59-14AM/oui/jlib/ojmisc.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/xml.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/srvm.jar://scratch/ohome/oracle/work/OraInstall2009-07-20_
04-59-14AM/oui/jlib/srvmasm.jar
oracle.sysman.oii.oiic.OiicInstaller
-scratchPath
//scratch/ohome/oracle/work/OraInstall2009-07-20_04-59-14AM
-sourceLoc
/scratch/oui/cd/Disk1/install/../stage/products.xml
```

```
                   -sourceType
                   network
                   -timestamp
                   2009-07-20_04-59-14AM
                   INFO: Environment Variables:

                   INFO: ORACLE_HOME = /scratch/ohome/oracle

                   INFO: PATH =
                   /usr/local/packages/ant/bin:/usr/local/packages/jdk15/bin:/usr/local/bin:/bin://
                   scratch/ohome/oracle/buildtools/bin://scratch/ohome/oracle/oui/bin:/scratch/ohome/
                   oracle/utl:/test:oratest:/pdp/pds/utl:/scratch/ohome/oracle/bin:/scratch/ohome/
                   oracle/nlsrtl3/bin:/opt/SUNWspro/bin:/usr/ccs/bin:/usr/kerberos/bin:/bin:/usr/bin:
                   /usr/local/bin:/usr/X11R6/bin
                   INFO: Oracle Universal Installer version is 11.2.0.1.0

                   INFO: Setting variable 'PREREQ_CONFIG_LOCATION' to ''. Received the value from
                   variable association.
                   INFO: Reader/Writer 'oracle.sysman.oii.oiic.OiicGlobalContextXMLReaderWriter' was
                   created.
                   INFO:
                   *** Welcome Page***
                   INFO: Initializing install inventory
                   INFO: Setting up install inventory
                   INFO:
                   *** Specify Source Location Page***
                   INFO: Initializing OUI Shiphome access setup
                   INFO: Loading global variables
                   INFO: Loading of globalvariables complete
                   INFO: DefaultifyLangiages : Value of oracle.installer.defaultifyLanguages : false
                   INFO: fetch the file: /scratch/oui/cd/Disk1/stage/shiphomeinfo.properties
                   INFO: Performing operation for OUI Shiphome access setup
                   INFO:
                   *** Select a Product to Install Page***
                   INFO: Setting the 'TopLevelComp ( ToplevelComp )' property to 'oracle.swd,
                   11.2.0.1.0, >0.0.0.0.0, [ 46 ][OH:2]'. Received the value from the default
                   settings.
                   INFO:
                   *** Select Installation Type Page***
                   INFO: Setting the 'InstallType ( DEP_MODE )' property to 'Complete'. Received the
                   value from the UI page.
                   INFO: Setting the 'TopLevelInstallType ( TLDepModes )' property to 'Complete'.
                   Received the value from the UI page.
                   INFO: The selected install type is "Complete".
                   INFO:
                   *** Specify Home Details Page***
                   INFO: Setting the 'OracleHome ( ORACLE_HOME )' property to '/scratch/homes/vc'.
                   Received the value from the UI page.
                   INFO: Setting the 'OracleHomeName ( ORACLE_HOME_NAME )' property to 'VC'. Received
                   the value from the UI page.
                   INFO: Initializing OUI Oracle Home access setup
                   INFO: Setting variable 'ORACLE_HOME' to '/scratch/homes/vc'. Received the value
                   from the UI page.
                   INFO: DefaultifyLangiages : Value of oracle.installer.defaultifyLanguages : false
                   INFO: Setting the PROD_HOME variable of 'oracle.swd_11.2.0.1.0' to ''. Received
                   the value from the variable calculation.
                   INFO: oracle.swd Product Location: Empty path specified.
                   INFO: Setting the PROD_HOME variable of 'oracle.swd.oil_11.2.0.1.0' to
                   '/scratch/homes/vc/osp/Libraries'. Received the value from the variable
                   calculation.
```

```
INFO: Setting the PROD_HOME variable of 'oracle.swd.oui_11.2.0.1.0' to
'/scratch/homes/vc/oui'. Received the value from the variable calculation.
INFO: Setting the PROD_HOME variable of 'oracle.swd.opatch_11.2.0.0.2' to
'/scratch/homes/vc/OPatch'. Received the value from the variable calculation.
INFO: Setting the PROD_HOME variable of 'oracle.swd.oui.core_11.2.0.1.0' to
'/scratch/homes/vc/oui'. Received the value from the variable calculation.
INFO: Setting the PROD_HOME variable of 'oracle.swd.jre_1.5.0.11.0' to
'/scratch/homes/vc/jre/1.5.0'. Received the value from the variable calculation.
INFO: Performing operation for OUI Oracle Home access setup
INFO: Unable to read /scratch/homes/vc/inventory/ContentsXML/comps.xml. Some
inventory information may be lost.
INFO:
*** Oracle Universal Installer Page***
INFO: Setting variable 'PROXY_USER' to ''. Received the value from a code block.
INFO: Setting variable 'PROXY_PWD' to '*Protected value, not to be logged*'.
Received the value from a code block.
INFO: Setting variable 'PROXY_HOST' to ''. Received the value from a code block.
INFO: Setting variable 'PROXY_PORT' to ''. Received the value from a code block.
INFO: Setting variable 'MYORACLESUPPORT_PASSWORD' to '*Protected value, not to be
logged*'. Received the value from a code block.
INFO: Setting variable 'METALINK_PASSWORD' to '*Protected value, not to be
logged*'. Received the value from a code block.
INFO: Setting the value of METALINK_PASSWORD with the value of MYORACLESUPPORT_
PASSWORD
INFO: Setting the value of SECURITY_UPDATES_VIA_METALINK
INFO: Setting variable 'SECURITY_UPDATES_VIA_METALINK' to 'false'. Received the
value from a code block.
INFO: Setting the value of SECURITY_UPDATES_VIA_METALINK with the value of
SECURITY_UPDATES_VIA_MYORACLESUPPORT
INFO: Setting variable 'COLLECTOR_RESPONSE_FILE' to ''. Received the value from a
code block.
INFO: Is the component oracle.sysman.ccr found in the Shiphome ? false
INFO: Is the component eligible to be installed in the home ? false
So setting the system property oracle.sysman.ccrinstall to false
INFO: The user has chosen to include the collector, but with the
shiphomeinfo.properties value, is the collector included for configuration ? :
false
INFO: Can we show the collector page ? false
INFO: Setting the value of METALINK_PASSWORD
INFO: Setting variable 'METALINK_PASSWORD' to '*Protected value, not to be
logged*'. Received the value from a code block.
INFO: Setting the value of SECURITY_UPDATES_VIA_METALINK
INFO: Setting variable 'SECURITY_UPDATES_VIA_METALINK' to 'false'. Received the
value from a code block.
INFO: Setting the value of METALINK_USERNAME
INFO: Setting the 'MyOracleSupportPassword ( MYORACLESUPPORT_PASSWORD )' property
to '*Protected value, not to be logged*'. Received the value from a code block.
INFO: Setting the 'ProxyUsername ( PROXY_USER )' property to ''. Received the
value from a code block.
INFO: Setting the 'ProxyPassword ( PROXY_PWD )' property to '*Protected value, not
to be logged*'. Received the value from a code block.
INFO: Setting the 'ProxyServer ( PROXY_HOST )' property to ''. Received the value
from a code block.
INFO: Setting the 'ProxyPort ( PROXY_PORT )' property to ''. Received the value
from a code block.
INFO: Setting the 'DeclineUpdates ( DECLINE_SECURITY_UPDATES )' property to
'false'. Received the value from the default settings.
INFO: Setting the 'EnableMyOracleSupportPassword ( SECURITY_UPDATES_VIA_
MYORACLESUPPORT )' property to 'false'. Received the value from the default
settings.
```

```
INFO: SRVM ClusterInfo.IsLocalOnly() called. Return value obtained is 'false'.
INFO: Clusterware location for 'oracle.crs 10.1.0.2.0' is not found.
INFO: Clusterware not found. Defaulting to vendor clusterware.
INFO: Vendor clusterware is not detected.
INFO: This is not a cluster system.
INFO: Node selection page will not be shown.
INFO: Defaulting prerequisite location to '/scratch/oui/cd/Disk1/stage/prereq'.
INFO:
*** Product-Specific Prerequisite Checks Page***
INFO: OUI-10185:Unable to copy
'/scratch/oui/cd/Disk1/stage/prereq/prerequisite.properties' to
'/scratch/homes/vc/inventory/prereqs/prerequisite.properties'. OUI-10186:The
source file '/scratch/oui/cd/Disk1/stage/prereq/prerequisite.properties' does not
exist.
INFO: No prerequisite checks specified for this installation.
INFO: Install type for "Oracle Installation Products 11.2.0.1.0 " is "Complete".
INFO: OUI-10066:Not all the dependencies for the component Oracle Installation
Products 11.2.0.1.0  could be found. Missing component oracle.swd.osp 11.2.0.1.0.
INFO: OUI-10066:Not all the dependencies for the component Oracle Installation
Products 11.2.0.1.0  could be found. Missing component oracle.swd.opatch
11.2.0.1.0.
INFO: Install type for "Oracle Installation Libraries 11.2.0.1.0 " is "Custom".
INFO: Install type for "Oracle Universal Installer 11.2.0.1.0 " is "Custom".
INFO: OUI-10066:Not all the dependencies for the component Oracle Universal
Installer 11.2.0.1.0  could be found. Missing component oracle.sysman.ccr
10.3.1.0.0.
INFO: Install type for "Oracle One-Off Patch Installer 11.2.0.0.2 " is "Custom".
INFO: Install type for "Installer SDK Component 11.2.0.1.0 " is "Custom".
INFO: Install type for "Java Runtime Environment 1.5.0.11.0 " is "Custom".
INFO: Setting the PRE_REQUISITE variable of 'oracle.swd_11.2.0.1.0' to 'true'.
Received the value from the variable calculation.
INFO: Setting the PRE_REQUISITE variable of 'oracle.swd.jre_1.5.0.11.0' to 'true'.
Received the value from the variable calculation.
INFO: Setting the PRE_REQUISITE variable of 'oracle.swd.oil_11.2.0.1.0' to 'true'.
Received the value from the variable calculation.
INFO: Setting the PRE_REQUISITE variable of 'oracle.swd.oui_11.2.0.1.0' to 'true'.
Received the value from the variable calculation.
INFO: Setting the PRE_REQUISITE variable of 'oracle.swd.oui.core_11.2.0.1.0' to
'true'. Received the value from the variable calculation.
INFO: Setting the PRE_REQUISITE variable of 'oracle.swd.opatch_11.2.0.0.2' to
'true'. Received the value from the variable calculation.
INFO:
*** Language Selection Page***
INFO: Setting the 'SelectedLanguages ( SELECTED_LANGUAGES )' property to 'en'.
Received the value from the default settings.
INFO:
*** Specify Local Host Name Page***
INFO: Setting the 'HostName ( ORACLE_HOSTNAME )' property to
'stacg22.us.oracle.com'. Received the value from the default settings.
INFO: *** Entering Component: oracle.swd installation
INFO: *** Entering Component: oracle.swd.oil installation
INFO: *** Entering Component: oracle.swd.oui installation
INFO: Calling Query generalQueries10.2.0.2.0  isCurrentPlatformInGroup
platGroup = WINDOWS
INFO: Query Returned: false
INFO: Setting variable 'isPlatGroupWindows' to 'false'. Received the value from
the variable calculation.
INFO: Setting variable 'JRE_MEMORY_OPTIONS' to ''. Received the value from the
variable calculation.
INFO: Setting variable 'JRE_LOCATION' to ''. Received the value from the variable
```

```
calculation.
INFO: Setting variable 'InstVer' to '11.2.0.1.0'. Received the value from the
variable calculation.
INFO: Setting variable 'ABSOLUTE_JRE_LOCATION' to ''. Received the value from the
variable calculation.
INFO: Setting variable 's_extLib' to
'/scratch/homes/vc/inventory/Scripts/ext/lib'. Received the value from the
variable calculation.
INFO: Setting variable 's_extJlib' to
'/scratch/homes/vc/inventory/Scripts/ext/jlib'. Received the value from the
variable calculation.
INFO: Setting variable 's_extBin' to
'/scratch/homes/vc/inventory/Scripts/ext/bin'. Received the value from the
variable calculation.
INFO: Setting variable 'OUIFolder' to 'null\Oracle Installation Products'.
Received the value from the variable calculation.
INFO: Setting variable 'InstLocProp' to '/scratch/homes/vc/oui'. Received the
value from the variable calculation.
INFO: *** Entering Component: oracle.swd.opatch installation
INFO: *** Entering Component: oracle.swd.oui.core installation
INFO: Setting variable 'InstVer' to '11.2.0.1.0'. Received the value from the
variable calculation.
INFO: *** Entering Component: oracle.swd.jre installation
INFO: oracle.srvm.formCluster:false
INFO:
*** Summary Page***
INFO:
-------------------------------------------------------------------------------
Summary
Global Settings
     Source: /scratch/oui/cd/Disk1/stage/products.xml
     Oracle Home: /scratch/homes/vc (VC)
     Installation Type: Complete
Product Languages
   English
Space Requirements
   / Required 151MB (includes 33MB temporary) : Available 5.36GB
New Installations (6 products)
   Oracle Installation Products 11.2.0.1.0
   Oracle Installation Libraries 11.2.0.1.0
   Oracle Universal Installer 11.2.0.1.0
   Oracle One-Off Patch Installer 11.2.0.0.2
   Installer SDK Component 11.2.0.1.0
   Java Runtime Environment 1.5.0.11.0
-------------------------------------------------------------------------------
INFO: The default response file generated with recorded values is null
INFO: -destinationFile option was not provided. Saving the default response file
as null
INFO: Recording the installation in file null.
INFO: Checkpoint:getting indexSession from checkpoint factory
INFO: Checkpoint:Index file :/scratch/homes/vc/inventory/checkpoints/oui/index.xml
not found.
INFO: Checkpoint:Initializing checkpointindex session in oiifwactionsphasewcde.
INFO: Checkpoint:Location is- /scratch/homes/vc/inventory
INFO: Setting checkpoint object
INFO: Method 'dispose()' Not implemented in class
'oracle.sysman.oii.oiif.oiifp.OiifpWizIntroPanel'
INFO: config-context initialized
INFO: *** Install Page***
INFO: Checkpoint:Initializing checkpoint session in oiifwinstallphasewcce.
```

Text files are also written in the `temp` directory:

```
<temp>\OraInstall\oraInstall<timestamp>.err
<temp>\OraInstall\oraInstall<timestamp>.out
```

If a problem occurs during the installation, you should look for messages with the SEVERE keyword and Java exceptions in these files.

Automated Inventory Backups

You can recover from a corrupt inventory or failed installations by using the following types of automated inventory backups:

- Central Inventory Backup
- Oracle Home Inventory Backup

Central Inventory Backup

The Central Inventory is automatically backed up whenever you execute the Oracle Universal Installer in any of the following modes:

- `- install` (new home)
- `- deinstall` (remove home)
- `- addNode`
- `- attachHome`
- `- detachHome`
- `- updateNodeList`

The backed up directory and file are:

Central Inventory Location/ContentsXML

The backup is located here:

Central Inventory Location/backup/time stamp/

Oracle Home Inventory Backup

The Oracle Home Inventory is automatically backed up on an existing Oracle home whenever you execute the Oracle Universal Installer in either of the following modes:

- `- install` (patchsets, upgrades, and one-offs)
- `- deinstall components` (does not remove the Oracle home)

The backed up directories and files are:

```
ORACLE_HOME/inventory/ContentsXML
ORACLE_HOME/inventory/Scripts
ORACLE_HOME/inventory/Clone
ORACLE_HOME/inventory/invDetails.properties
```

The backup is located here:

```
ORACLE_HOME/inventory//backup/time stamp/
```

Oracle Universal Installer Errors

Refer to the release notes for information on any Oracle Universal Installer limitations for a particular version.

The most common Oracle Universal Installer errors are listed below:

- Out of "temp" space

  Make sure you have enough space in the default `TEMP` or `TMP` directory. Otherwise, the installation will not succeed. The amount of space required depends on the product being installed.

  > **Note:** On UNIX and Windows, Oracle Universal Installer looks for `%TEMP%` then `%TMP%`. If neither is set, Oracle Universal Installer will default to `/tmp` on UNIX and `c:\temp` on Windows.

- Incomplete stage and missing files in file groups

  Make sure you have a good staging area. You may have incorrectly copied the staging area from the CD to the hard disk.

- Exceptions from action libraries:
  - NT Services, if a service is already started
  - File permissions

- `UNCAUGHT_RUNTIME` exception when you try to install the product on your machine

  This is a blanket error catch that catches any run-time errors like `NullPointers` from libraries. Check if the `temp\OraInstall\<timestamp>.err` has any stack trace. You will need to call support to debug the staging area.

Other Tips

The following tips may help you to troubleshoot problems:

- Make sure the correct version of the JRE is specified in the `PATH`.

- You can use the following debugging flags in the runInstaller command:
  - **-printdiskusage**

    Logs debug information for disk usage.
  - **-printmemory**

    Logs debug information for memory usage.
  - **-printtime**

    Logs debug information for time usage.

- If Java cannot run applications from a Sun machine on an NCD X terminal (Oracle Universal Installer does not start), remove the following file:

  `$JAVA_HOME/lib/font.properties`

- If you are deleting an Oracle home manually without using Oracle Universal Installer, the products remain registered with Oracle Universal Installer. You must then simulate a de-install so that all Oracle Universal Installer references are removed.

■ If the installation is canceled:

– Products will not be registered with Oracle Universal Installer.

– Some files may have been copied, depending on when you issued the cancellation.

X terminal emulators that were tested with Oracle Universal Installer are listed in the following sections.

Exceed

There is a known compatibility issue that Hummingbird has identified to be a problem with Exceed. You can fix it by going into XConfig/Screen Definition/Screen 0 and changing Window Manager from "Default to Native" to "Native." See Figure B–1 to see the proper setup.

Figure B–1   Setting Exceed's Window Manager to "Native"
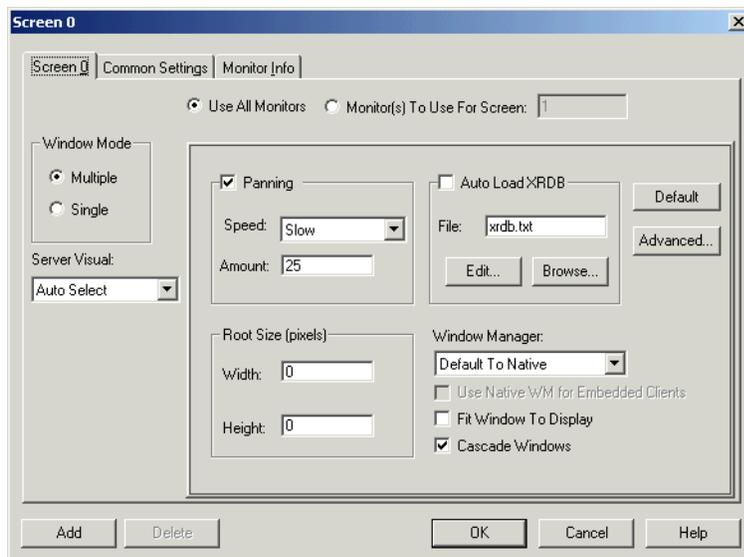
Another known issue where Exceed fails to recognize fonts is documented at the following site:

```
http://developer.java.sun.com/developer/bugParade/bugs/4262023.html
```

Reflection X

■ Refection X 6.2+: Select Settings/Window Manager/ Default Local Window Manager/ Reflection Window Manager. You may also have to select "X terminal desktop."

■ Reflection X 6.0: Select Connection/New XDMCP Connection/Connection Settings/Direct method/ type host destination and click **Connect.**

Tarantella

There is a known problem when using Tarantella to send DISPLAY to your Windows system, for instance, when using hosted UNIX systems. The bottoms of some Oracle Universal Installer screens may appear truncated, with buttons not visible. The problem occurs only when you use an individual Tarantella terminal window so that

the Oracle Universal Installer window appears alone without a broader X-windowing environment.

To work around this problem, do one of the following:

- Use keyboard shortcuts to invoke these commands:

  – Use **Alt-N** for **Next** to move to the next screen.

  – On the summary page, use **Alt-I** to start the installation.

- Use the Window Manager window in Tarantella, which displays the entire screen as if it were a UNIX monitor. Installer dialogs will appear properly in Window Manager.

C

Understanding Oracle Universal Installer Commands

The Oracle Universal Installer is run with various options and command-line arguments. This appendix lists the syntax of Oracle Universal Installer and the various command-line options available in Oracle Universal Installer. The following command shows the syntax for running the Oracle Universal Installer:

```
./runInstaller or setup.exe <option> [-command_line_variable=value]
```

In the preceding command, the following variables are used:

- *option*: The options available in Oracle Universal Installer. They are described in Table C–1, " Options in Oracle Universal Installer".

- *command_line_variable*: The command-line variable for the option. They are described in Table C–2, " Command Line Variables in Oracle Universal Installer".

> **Note:** You can pass the command line variable in either of the following three ways:
>
> 1. *<command_line_variable>={"value1","value2"}*
>
> 2. *"<command_line_variable>={value1,value2}"*
>
> 3. <command_line_variable>="{value1,value2}"

Options Available in Oracle Universal Installer

Table C–1 lists the various options available in Oracle Universal Installer.

Table C–1   Options in Oracle Universal Installer

Option	Description
-clusterware oracle.crs,<crs version>	Version of the Oracle Clusterware installed.
-crsLocation <Path>	Specifies the path to the Oracle Clusterware home location. This flag is used only for cluster installations. Specifying this flag overrides the Oracle Clusterware information obtained from the Central Inventory.
-invPtrLoc <full path of oraInst.loc>	Points to a different inventory location. The `orainst.loc` file contains the following entries: `inventory_loc=<location of central inventory>inst_group=<>` This flag can only be passed on UNIX platforms.

Table C–1   (Cont.) Options in Oracle Universal Installer

Option	Description
-jreLoc <location>	Path where the Java Runtime Environment is installed. You need to mention this path to override the default path. This option does not support relative paths, so you need to specify the absolute path.
-logLevel <level>	Filters log messages that have a lesser priority level than the level specified. Valid options are: severe, warning, info, config, fine, finer, and finest.
-paramFile <location of file>	Specifies the location of the oraparam.ini file to be used by Oracle Universal Installer.
-responseFile <Path>	Specifies the location of the response file to use. This option is used with the -silent flag.
-sourceLoc <location of products.xml>	Specifies the shiphome location.
-addLangs	Adds new languages to an already installed product.
-addNode	Adds node(s) to the installation. You cannot use this flag in silent mode with response files. You can pass the required session variable through the command line. You must pass the CLUSTER_NEW_NODES, CLUSTER_NEW_PRIVATE_NODE_NAMES, and CLUSTER_NEW_VIRTUAL_HOSTNAMES session variables when using this flag.
-attachHome	Attaches homes to the Oracle Universal Installer inventory. You must pass the ORACLE_HOME session variable when using this flag.
-cfs	Indicates that the Oracle home specified is on a cluster file system (shared). This is mandatory when '-local' flag is specified so that Oracle Universal Installer can register the home appropriately into the inventory.
-clone	Used for cloning an Oracle home from a source location to a target location. You must pass the ORACLE_HOME session variable when using this flag.
-debug	Used for getting the debug information from Oracle Universal Installer.
-detachHome	Detaches homes from the Oracle Universal Installer inventory without deleting the inventory directory inside the Oracle home. You must pass the ORACLE_HOME session variable when using this flag.
-enableRollingUpgrade	Used in cluster environments to enable an upgrade of a product on a subset of nodes (on which the product was installed).
-executeSysPrereqs	Executes system prerequisite checks and exits.
-force	Enables silent mode installation on a non-empty directory without warning.
-help	Lists the syntax and help information.
-ignorePatchConflicts	Ignores all conflicts with existing interim patches during an upgrade. The conflicting interim patches are removed from the home.
-ignoreSysPrereqs	Ignores the results of the system prerequisite checks.
-local	Performs the specified operation on the local node irrespective of the cluster nodes specified.
-printdiskusage	Logs debug information for disk usage.

Table C–1   (Cont.) Options in Oracle Universal Installer

Option	Description
-printmemory	Logs debug information for memory usage.
-printtime	Logs debug information for time usage.
-removeAllPatches	Removes all interim patches from the home.
-silent	Used for silent mode operations. The input can be a response file or a list of command line variable value pairs. You can use the -responsefile flag along with this flag.
-updateNodeList	Updates the node list for this home in the Oracle Universal Installer inventory. You can pass the ORACLE_HOME, LOCAL_NODE, and CLUSTER_NODES session variable when using this flag
-waitforcompletion	The command console will wait for Oracle Universal Installer to exit if you specify this flag.
-nobackground	Specifies to Oracle Universal Installer not to show background images.
-noclusterEnabled	Indicates that no cluster nodes are specified.
-noconsole	Suppresses the display of messages to the console.
-nowait	Used on a Windows platform. Specifies not to wait for the user to press Enter on the console after the task (installation, etc.) is complete.
-formCluster	Installs the Oracle Clusterware in order to form the cluster.
-remotecp <Path>	This flag can only be passed on UNIX platforms. This is used only for cluster installs. This flag specifies the path to the remote copy program on the local cluster node. For example, the path for scp is /usr/bin/scp.
-remoteshell <Path>	This flag can only be passed on UNIX platforms. This is used only for cluster installs. This specifies the path to the remote shell program on the local cluster node. For example, the path for ssh is /usr/bin/ssh.

Command-line Variables Available in Oracle Universal Installer

Table C–2 lists the command-line variables available in Oracle Universal Installer:

Table C–2   Command Line Variables in Oracle Universal Installer

Command Line Variable	Description
INVENTORY_ LOCATION	Specifies the inventory location.
ORACLE_HOME	Specifies the ORACLE_HOME location.
ORACLE_HOME_ KEY	Specifies the ORACLE HOME key name. This is used only on Windows platform.
ORACLE_HOME_ FOLDER	Specifies the ORACLE HOME folder name. This is used only on Windows platform.
ORACLE_HOME_ SERVICE	Specifies the ORACLE HOME service name. This is used only on Windows platform.
ANONYMOUS_ LOGIN	Specifies an anonymous login.

Table C–2 (Cont.) Command Line Variables in Oracle Universal Installer

Command Line Variable	Description
LOGIN_NAME	Specifies the login name.
PASSWORD	Specifies the password.
LOGIN_CATEGORY	Specifies the login category.
UNIX_GROUP_ NAME	Specifies the group name on a UNIX platform.
FROM_LOCATION	Specifies the location from where the component gets installed. This is the complete path to the products.xml file.
FROM_LOCATION_ CD_LABEL	Specifies the label of the CD where the products.xml file resides in a multi-CD installation. The label can be found in the file disk label in the directory where the products.xml resides.
COMPONENT_ LANGUAGES	Specifies the languages in which the components will be installed.
SELECTED_ LANGUAGES	Specifies the selected languages.
SELECTED_ PLATFORMS	Specifies the selected platforms.
TOPLEVEL_ COMPONENT	Specifies the top level component to be installed in the current session.
ToplevelComp	Specifies the top level component id.
INSTALL_TYPE	Specifies the installation type of the components.
NEXT_SESSION	Specifies if you want users to go back to the File Locations page for another installation. This flag also needs to be set to true in order to process another response file (see NEXT_SESSION_RESPONSE). This argument is used for chain installs.
NEXT_SESSION_ ON_FAIL	Specifies the next session response. Set to true to enable users to invoke another session even if the current installation session has failed. This flag is only relevant if NEXT_SESSION is set to true. This argument is used for chain installs.
NEXT_SESSION_ RESPONSE	Specifies the full path of the next session's response file. If only a file name is specified, the response file is retrieved from the <TEMP>/oraInstall directory. This variable is active only if NEXT_ SESSION is set to true. This argument is used for chain installs.
LOCATION_FOR_ DISK	Name used to specify the complete path to the other disks.
CLUSTER_NODES	Specifies the cluster node names selected by the user for installation.
CLUSTER_NEW_ NODES	Specifies the new cluster nodes to be used for node addition.
REMOTE_NODES	Specifies the remote nodes.
REMOTE_CLEAN_ MACHINES	Specifies the remote nodes which do not have an installation inventory set up.
LOCAL_NODE	Specifies the local node.
EXISTING_REMOTE_ NODES	Specifies the remote nodes on which the current home is already installed.
CRS	Specifies whether the Oracle home is an Oracle Clusterware home or not.

Table C–2   (Cont.) Command Line Variables in Oracle Universal Installer

Command Line Variable	Description
CLUSTER_NEW_ PRIVATE_NODE_ NAMES	Specifies the private node names for the new nodes that are used for node addition.
CLUSTER_NEW_ HOST_NAMES	Specifies the host names for the new cluster nodes.
CLUSTER_NEW_ NODE_NUMBERS	Specifies the node numbers of the new cluster nodes
CLUSTER_NEW_ VIRTUAL_ HOSTNAMES	Specifies the virtual host names for the new cluster nodes.
ACCEPT_LICENSE_ AGREEMENT	Specifies the license agreement status. By setting this variable to true, you are accepting the license agreement.
REGISTRATION_ KEYWORD	Specifies the registration keyword value. This can be obtained from the URL specified in the Product Registration Page.
RESTART_SYSTEM	Specifies whether to restart the system or not. Set to true to enable automatic restart of the system. If set to false, the installer exits without restarting. No exit confirmation dialog is shown.
RESTART_REMOTE_ SYSTEM	Specifies whether to restart the remote system or not. Set to true to enable automatic restarting of the remote system. If set to false, the installer does not restart remote systems. No exit confirmation dialog is shown.
SESSION_MODE	This session variable holds the current Oracle Universal Installer execution mode; that is, "Install" for an install and/or de-install session, or "Clone" for a cloning (home fix-up) session.
LOG_LOCATION	Specifies the log file location. This variable specifies the directory that contains the log files generated during installation. This directory is located within the Oracle home.
IGNORE_PREREQ	Specifies whether to skip the prerequisite checks or not. Set to true to continue component installation even when the prerequisite check fails.
ORACLE_ HOSTNAME	Specifies the host name of the system.
PREREQ_CONFIG_ LOCATION	Specifies the location where prerequisite configuration files are available.
PREREQ_LOG_ LOCATION	Specifies the prereqs log location where prerequisite results will be generated.
PREREQ_ENTRY_ POINT	Specifies the log location where prerequisite results will be generated.
ROOTSH_STATUS	Specifies the root.sh status — whether root.sh is required/before config or after config. This flag is used only on UNIX platforms.
ROOTSH_ LOCATION	Specifies the Root.sh location. By default it is set to OH\root.sh. This flag is used only on UNIX platforms.
USE_PREREQ_ CHECKER	Specifies if the prereq checker is to be used during the installation or not. The value for this variable is set based on the value in the oraparam.ini file.
USE_OLD_INSTALL_ PREREQS	Specifies if old style system prereqs are to be enabled in the installation scripts. The PRE_REQUISITE code should use this variable to determine which checks can be turned on or off. The value of this variable is set based on the value in the Oraparam.ini.

Table C–2   (Cont.) Command Line Variables in Oracle Universal Installer

Command Line Variable	Description
NEXT_SESSION_ PROGRESS_TITLE	Specifies the title of the progress dialog shown during the setup of the next session. This flag is used in chain installs.
NEXT_SESSION_ PROGRESS_TEXT	Specifies the text above the progress bar in the progress dialog shown during the setup of next session. This flag is used in chain installs.

D

Sample Files

This appendix provides the following sample files:

- Sample Response File
- Sample ORAPARAM.INI File
- Sample Components File

Sample Response File

Response files are of type `.rsp` and are used by Oracle Universal Installer to run silent installations.

```
####################################################################
## Copyright(c) 1999, 2009 Oracle. All rights reserved.          ##
##                                                               ##
## Specify values for the variables listed below to customize    ##
## your installation.                                            ##
##                                                               ##
## Each variable is associated with a comment. The comment       ##
## identifies the variable type.                                 ##
##                                                               ##
## Please specify the values in the following format:            ##
##                                                               ##
##          Type         Example                                 ##
##          String       "Sample Value"                          ##
##          Boolean      True or False                           ##
##          Number       1000                                    ##
##          StringList   {"String value 1","String Value 2"}     ##
##                                                               ##
## The values that are given as <Value Required> need to be      ##
## specified for a silent installation to be successful.         ##
##                                                               ##
##                                                               ##
## This response file is generated by Oracle Software            ##
## Packager.                                                     ##
####################################################################

RESPONSEFILE_VERSION=2.2.1.0.0


#-------------------------------------------------------------------------------
#Name      : UNIX_GROUP_NAME
#Datatype  : String
#Description: Unix group to be set for the inventory directory. Valid only in Unix
platforms.
#Example: UNIX_GROUP_NAME = "install"
```

```
#-------------------------------------------------------------------------------
UNIX_GROUP_NAME=<Value Unspecified>


#-------------------------------------------------------------------------------
#Name       : FROM_LOCATION
#Datatype   : String
#Description: Complete path to the products.xml.
#Example: FROM_LOCATION = "../stage/products.xml"
#-------------------------------------------------------------------------------
FROM_LOCATION="../stage/products.xml"


#-------------------------------------------------------------------------------
#Name       : FROM_LOCATION_CD_LABEL
#Datatype   : String
#Description: This variable should only be used in multi-CD installations. It
includes the #label of the compact disk where the file "products.xml" exists. The
label can be found in #the file "disk.label" in the same directory as
products.xml.

#Example: FROM_LOCATION_CD_LABEL = "CD Label"
#-------------------------------------------------------------------------------
FROM_LOCATION_CD_LABEL="LABEL1"


#-------------------------------------------------------------------------------
#Name       : ORACLE_HOME
#Datatype   : String
#Description: Complete path of the Oracle Home.
#Example: ORACLE_HOME = "C:\OHOME1"
#-------------------------------------------------------------------------------
ORACLE_HOME=<Value Required>


#-------------------------------------------------------------------------------
#Name       : ORACLE_BASE
#Datatype   : String
#Description: Complete path of the Oracle Base.
#Example: ORACLE_BASE =
#-------------------------------------------------------------------------------
ORACLE_BASE=<Value Required>


#-------------------------------------------------------------------------------
#Name       : ORACLE_HOME_NAME
#Datatype   : String
#Description: Oracle Home Name. Used in creating folders and services.
#Example: ORACLE_HOME_NAME = "OHOME1"
#-------------------------------------------------------------------------------
ORACLE_HOME_NAME="OHOME1"


#-------------------------------------------------------------------------------
#Name       : SUPPRESS_BUGLIST_WARNING
#Datatype   : Boolean
#Description: Set to true if the Welcome page in OUI needs to be shown.
#Example: SUPPRESS_BUGLIST_WARNING = false
#-------------------------------------------------------------------------------
SUPPRESS_BUGLIST_WARNING=false


#-------------------------------------------------------------------------------
#Name       : NEXT_SESSION
#Datatype   : Boolean
#Description: Set to true to allow users to go back to the File Locations page for
```

another #installation. This flag also needs to be set to true in order to process
another response #file (see NEXT_SESSION_RESPONSE).

```
#Example: NEXT_SESSION = true
#--------------------------------------------------------------------------------
NEXT_SESSION=true


#--------------------------------------------------------------------------------
#Name       : NEXT_SESSION_ON_FAIL
#Datatype   : Boolean
#Description: Set to true to allow users to invoke another session even if current
install #session has failed. This flag is only relevant if NEXT_SESSION is set to
true.

#Example: NEXT_SESSION_ON_FAIL = true
#--------------------------------------------------------------------------------
NEXT_SESSION_ON_FAIL=true


#--------------------------------------------------------------------------------
#Name       : NEXT_SESSION_RESPONSE
#Datatype   : String
#Description: Set to true to allow users to go back to the File Locations page for
another #installation. This flag also needs to be set to true in order to process
another response #file (see NEXT_SESSION_RESPONSE).
#Example: NEXT_SESSION_RESPONSE = "nextinstall.rsp"
#--------------------------------------------------------------------------------
NEXT_SESSION_RESPONSE=<Value Unspecified>


#--------------------------------------------------------------------------------
#Name       : DEINSTALL_LIST
#Datatype   : StringList
#Description: List of components to be deinstalled during a deinstall session.
#The following choices are available. The value should contain only one of these
choices.
#The choices are of the form Internal Name, Version : External name. Please use
the internal #name and version while specifying the value.

#    oracle.swd, 11.2.0.1.0 : Oracle Installation Products 11.2.0.1.0
#Example: DEINSTALL_LIST = {"oracle.swd","11.2.0.1.0"}
#--------------------------------------------------------------------------------
DEINSTALL_LIST={"oracle.swd","11.2.0.1.0"}


#--------------------------------------------------------------------------------
#Name       : CLUSTER_NODES
#Datatype   : StringList
#Description: This variable represents the cluster node names selected by the user
for #installation.
#Example: CLUSTER_NODES = {"node1"}
#--------------------------------------------------------------------------------
CLUSTER_NODES=<Value Unspecified>


#--------------------------------------------------------------------------------
#Name       : REMOTE_NODES
#Datatype   : StringList
#Description: This variable represents the remote node names on which installation
is carried #out.
#Example: REMOTE_NODES =
#--------------------------------------------------------------------------------
REMOTE_NODES=<Value Required>
```

```
#--------------------------------------------------------------------------------
#Name       : LOCAL_NODE
#Datatype   : String
#Description: This variable represents the local node.
#Example: LOCAL_NODE =
#--------------------------------------------------------------------------------
LOCAL_NODE=<Value Required>


#--------------------------------------------------------------------------------
#Name       : RESTART_SYSTEM
#Datatype   : Boolean
#Description: Set to true to allow automatic restart of the system, if set to
false then #installer will exit without restarting, no exit confirmation dialog is
shown.

#Example: RESTART_SYSTEM = false
#--------------------------------------------------------------------------------
RESTART_SYSTEM=<Value Unspecified>


#--------------------------------------------------------------------------------
#Name       : RESTART_REMOTE_SYSTEM
#Datatype   : Boolean
#Description: Set to true to allow automatic restart of the remote systems, if set
to false,  #installer will not restart the remote systems, no exit confirmation
dialog is shown.

#Example: RESTART_REMOTE_SYSTEM = false
#--------------------------------------------------------------------------------
RESTART_REMOTE_SYSTEM=<Value Unspecified>


#--------------------------------------------------------------------------------
#Name       : ORACLE_HOSTNAME
#Datatype   : String
#Description: This variable holds the hostname of the system as set by the user.
#Example: ORACLE_HOSTNAME =
#--------------------------------------------------------------------------------
ORACLE_HOSTNAME=<Value Unspecified>


#--------------------------------------------------------------------------------
#Name       : REMOVE_HOMES
#Datatype   : StringList
#Description: List of the homes to be removed during a deinstall session. Each
home is #represented by its full path.

#Example: REMOVE_HOMES = {<full_path_of_home1>,<full_path_of_home2>, ...}
#--------------------------------------------------------------------------------
REMOVE_HOMES=<Value Unspecified>


#--------------------------------------------------------------------------------
#Name       : ACCEPT_LICENSE_AGREEMENT
#Datatype   : Boolean
#Description: By setting this variable to true, you are accepting the license
agreement. This #variable is used only for silent installations.

#Example: ACCEPT_LICENSE_AGREEMENT = true
#--------------------------------------------------------------------------------
ACCEPT_LICENSE_AGREEMENT=true


#--------------------------------------------------------------------------------
#Name       : METALINK_LOCATION
```

```
#Datatype   : String
#Description: This variable represents the Oracle metalink location.
#Example: METALINK_LOCATION =
#--------------------------------------------------------------------------------
METALINK_LOCATION=<Value Required>


#--------------------------------------------------------------------------------
#Name       : METALINK_USERNAME
#Datatype   : String
#Description: This variable represents the Oracle metalink user name.
#Example: METALINK_USERNAME =
#--------------------------------------------------------------------------------
METALINK_USERNAME=<Value Required>


#--------------------------------------------------------------------------------
#Name       : MYORACLESUPPORT_USERNAME
#Datatype   : String
#Description: This variable represents the Oracle metalink user name.
#Example: MYORACLESUPPORT_USERNAME =
#--------------------------------------------------------------------------------
MYORACLESUPPORT_USERNAME=<Value Required>


#--------------------------------------------------------------------------------
#Name       : METALINK_PASSWORD
#Datatype   : String
#Description: This variable represents the corresponding Oracle metalink password.
#Example: METALINK_PASSWORD =
#--------------------------------------------------------------------------------
METALINK_PASSWORD=<Value Required>


#--------------------------------------------------------------------------------
#Name       : MYORACLESUPPORT_PASSWORD
#Datatype   : String
#Description: This variable represents the corresponding Oracle metalink password.
#Example: MYORACLESUPPORT_PASSWORD =
#--------------------------------------------------------------------------------
MYORACLESUPPORT_PASSWORD=<Value Required>


#--------------------------------------------------------------------------------
#Name       : PROXY_HOST
#Datatype   : String
#Description: The proxy host  used to connect to Oracle metalink.
#Example: PROXY_HOST =
#--------------------------------------------------------------------------------
PROXY_HOST=<Value Required>


#--------------------------------------------------------------------------------
#Name       : PROXY_PORT
#Datatype   : String
#Description: The proxy port used to connect to Oracle metalink.
#Example: PROXY_PORT =
#--------------------------------------------------------------------------------
PROXY_PORT=<Value Required>


#--------------------------------------------------------------------------------
#Name       : PROXY_REALM
#Datatype   : String
#Description: The realm for the proxy used to connect to Oracle metalink.
#Example: PROXY_REALM =
#--------------------------------------------------------------------------------
```

```
                PROXY_REALM=<Value Required>

                #--------------------------------------------------------------------------------
                #Name      : PROXY_USER
                #Datatype  : String
                #Description: The username for the proxy  used to connect to Oracle metalink.
                #Example: PROXY_USER =
                #--------------------------------------------------------------------------------
                PROXY_USER=<Value Required>

                #--------------------------------------------------------------------------------
                #Name      : PROXY_PWD
                #Datatype  : String
                #Description: The password for the proxy  used to connect to Oracle metalink.
                #Example: PROXY_PWD =
                #--------------------------------------------------------------------------------
                PROXY_PWD=<Value Required>
                #--------------------------------------------------------------------------------
                #Name      : DONT_PROXY_FOR
                #Datatype  : String
                #Description: The dont proxy for list.
                #Example: DONT_PROXY_FOR =
                #--------------------------------------------------------------------------------
                DONT_PROXY_FOR=<Value Required>

                #--------------------------------------------------------------------------------
                #Name      : DECLINE_SECURITY_UPDATES
                #Datatype  : Boolean
                #Description: OUI Session variable set to decline from receiving the security
                updates
                #Example: DECLINE_SECURITY_UPDATES =
                #--------------------------------------------------------------------------------
                DECLINE_SECURITY_UPDATES=<Value Required>

                #--------------------------------------------------------------------------------
                #Name      : COLLECTOR_RESPONSE_FILE
                #Datatype  : String
                #Description: OUI Session variable used to provide the OCM response file location
                #Example: COLLECTOR_RESPONSE_FILE =
                #--------------------------------------------------------------------------------
                COLLECTOR_RESPONSE_FILE=<Value Required>

                #--------------------------------------------------------------------------------
                #Name      : SECURITY_UPDATES_VIA_METALINK
                #Datatype  : Boolean
                #Description: OUI Session variable used to set if the security updates should be
                received via #Metalink details
                #Example: SECURITY_UPDATES_VIA_METALINK =
                #--------------------------------------------------------------------------------
                SECURITY_UPDATES_VIA_METALINK=<Value Required>

                #--------------------------------------------------------------------------------
                #Name      : SECURITY_UPDATES_VIA_MYORACLESUPPORT
                #Datatype  : Boolean
                #Description: OUI Session variable used to set if the security updates should be
                received via #Metalink details
                #Example: SECURITY_UPDATES_VIA_MYORACLESUPPORT =
                #--------------------------------------------------------------------------------
                SECURITY_UPDATES_VIA_MYORACLESUPPORT=<Value Required>
```

```
#-------------------------------------------------------------------------------
#Name       : TOPLEVEL_COMPONENT
#Datatype   : StringList
#Description: The top level component to be installed in the current session.
#The following choices are available. The value should contain only one of these
choices.
#The choices are of the form Internal Name, Version : External name. Please use
the internal #name and version while specifying the value.
#    oracle.swd, 11.2.0.1.0 : Oracle Installation Products 11.2.0.1.0
#Example: TOPLEVEL_COMPONENT = {"oracle.swd","11.2.0.1.0"}
#-------------------------------------------------------------------------------
TOPLEVEL_COMPONENT={"oracle.swd","11.2.0.1.0"}




#-------------------------------------------------------------------------------
#Name       : SELECTED_LANGUAGES
#Datatype   : StringList
#Description: Languages in which the components will be installed.
#The following choices are available. The value should contain only one of these
choices.
#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.
#    en,    : English
#    fr,    : French
#    ar,    : Arabic
#    bn,    : Bengali
#    pt_BR,    : Brazilian Portuguese
#    bg,    : Bulgarian
#    fr_CA,    : Canadian French
#    ca,    : Catalan
#    hr,    : Croatian
#    cs,    : Czech
#    da,    : Danish
#    nl,    : Dutch
#    ar_EG,    : Egyptian
#    en_GB,    : English (United Kingdom)
#    et,    : Estonian
#    fi,    : Finnish
#    de,    : German
#    el,    : Greek
#    iw,    : Hebrew
#    hu,    : Hungarian
#    is,    : Icelandic
#    in,    : Indonesian
#    it,    : Italian
#    ja,    : Japanese
#    ko,    : Korean
#    es,    : Latin American Spanish
#    lv,    : Latvian
#    lt,    : Lithuanian
#    ms,    : Malay
#    es_MX,    : Mexican Spanish
#    no,    : Norwegian
#    pl,    : Polish
#    pt,    : Portuguese
#    ro,    : Romanian
#    ru,    : Russian
#    zh_CN,    : Simplified Chinese
#    sk,    : Slovak
```

```
#     sl,    : Slovenian
#     es_ES,   : Spanish
#     sv,    : Swedish
#     th,    : Thai
#     zh_TW,   : Traditional Chinese
#     tr,    : Turkish
#     uk,    : Ukrainian
#     vi,    : Vietnamese
#Example: SELECTED_LANGUAGES = {"en"}
#-------------------------------------------------------------------------------
#SELECTED_LANGUAGES={"en"}


#-------------------------------------------------------------------------------
#Name       : COMPONENT_LANGUAGES
#Datatype   : StringList
#Description: Languages in which the components will be installed.
#The following choices are available. The value should contain only one of these
choices.
#The choices are of the form Internal Name : External name. Please use the
internal name while specifying the value.

#     en,    : English
#     fr,    : French
#     ar,    : Arabic
#     bn,    : Bengali
#     pt_BR,   : Brazilian Portuguese
#     bg,    : Bulgarian
#     fr_CA,   : Canadian French
#     ca,    : Catalan
#     hr,    : Croatian
#     cs,    : Czech
#     da,    : Danish
#     nl,    : Dutch
#     ar_EG,   : Egyptian
#     en_GB,   : English (United Kingdom)
#     et,    : Estonian
#     fi,    : Finnish
#     de,    : German
#     el,    : Greek
#     iw,    : Hebrew
#     hu,    : Hungarian
#     is,    : Icelandic
#     in,    : Indonesian
#     it,    : Italian
#     ja,    : Japanese
#     ko,    : Korean
#     es,    : Latin American Spanish
#     lv,    : Latvian
#     lt,    : Lithuanian
#     ms,    : Malay
#     es_MX,   : Mexican Spanish
#     no,    : Norwegian
#     pl,    : Polish
#     pt,    : Portuguese
#     ro,    : Romanian
#     ru,    : Russian
#     zh_CN,   : Simplified Chinese
#     sk,    : Slovak
#     sl,    : Slovenian
#     es_ES,   : Spanish
```

```
#     sv,    : Swedish
#     th,    : Thai
#     zh_TW,   : Traditional Chinese
#     tr,    : Turkish
#     uk,    : Ukrainian
#     vi,    : Vietnamese
#Example: COMPONENT_LANGUAGES = {"en"}
#Component  : oracle.swd
#------------------------------------------------------------------------------

COMPONENT_LANGUAGES={"en"}


#------------------------------------------------------------------------------
#Name       : INSTALL_TYPE
#Datatype   : String
#Description: Installation type of the component.
#The following choices are available. The value should contain only one of these
choices.
#The choices are of the form Internal Name : External name. Please use the
internal name #while specifying the value.

#    Complete,   : Complete
#    Typical,   : Minimum
#    Custom,   : Custom
#Example: INSTALL_TYPE = "Complete"
#Component  : oracle.swd
#------------------------------------------------------------------------------

INSTALL_TYPE="Complete"
```

Sample ORAPARAM.INI File

The `oraparam.ini` file is Oracle Universal Installer's initialization file. It should be located in the same directory as the Oracle Universal Installer executable file (`setup.exe` or `runInstaller.sh`).

```
[Oracle]
DISTRIBUTION=TRUE
SOURCE=../stage/products.xml
LICENSE_LOCATION=
JRE_LOCATION=../stage/Components/oracle.swd.jre/1.4.2.0.0/1/DataFiles
JRE_MEMORY_OPTIONS=" -mx150m"
DEFAULT_HOME_LOCATION=oracle/product/11.2.0/db
DEFAULT_HOME_NAME=OraDb11g_home
NO_BROWSE=/net
NLS_ENABLED=TRUE
BOOTSTRAP=TRUE
PREREQ_CONFIG_LOCATION=../stage/prereq
OUI_VERSION=11.2.0.0.0
#THIN_JDBC_FILENAME is optional and defaults to classes12.jar
#The value specified for this should be packaged with OUI, and should
#be relative to <OUI expanded stagedir>/jlib/
THIN_JDBC_FILENAME=classes12.jar
#JRE_OSDPARAM is to set OS dependent param for JRE ( mainly for native VM in
1.3.1)
#JRE_OSDPARAM is optional and should be set to -native for the JRE's
#that support native VM ( mainly for Unix platforms ), in JRE 1.3.1
#For JRE 1.4.1 this should be set to empty or the type of VM that is
#supported client/server. The default value is -native in UNIX platforms
#that supports native VM
```

```
#Unix supporting native - JRE_OSDPARAM="-native"
#Unix NOT supporting native and 1.4.1 - JRE_OSDPARAM=""
JRE_OSDPARAM=""
CLUSTERWARE={"oracle.crs","11.2.0.0.0"}
#RUN_OUICA specifies the batch script name that needs to be run
#The script is ouica.bat for win32, and ouica.sh for solaris.
#If the value is not specified, then the OUICA script is not run
RUN_OUICA=ouica.sh

[Certified Versions]
Linux=redhat-2.1,UnitedLinux-1.0,redhat-3,SuSE-9

[Linux-redhat-2.1-optional]
```

Sample Components File

The components file contains details of all the components as well as patchsets or interim patches installed in the Oracle home.

```
<?xml version="1.0" standalone="yes" ?>
<!-- Copyright (c) 1999, 2009, Oracle. All rights reserved. -->
<!-- Do not modify the contents of this file by hand. -->
<PRD_LIST>
<TL_LIST>
<COMP NAME="oracle.server" VER="11.2.0.1.0" BUILD_NUMBER="0" REP_VER="0.0.0.0.0"
RELEASE="Production" INV_LOC="Components/oracle.server/11.2.0.1.0/1/" LANGS="en"
XML_INV_LOC="Components21/oracle.server/11.2.0.1.0/" ACT_INST_VER="11.2.0.1.0"
DEINST_VER="11.2.0.0.0" INSTALL_TIME="2009.Jul.21 11:00:34 PDT" INST_
LOC="/scratch/rv/dbbase/dbhome_1/oracle.server">
    <EXT_NAME>Oracle Database 11g</EXT_NAME>
    <DESC>Installs an optional preconfigured starter database, product options,
management tools, networking services, utilities, and basic client software for an
Oracle Database server. This option also supports Automatic Storage Management
database configuration.</DESC>
    <DESCID>COMPONENT_DESC</DESCID>
    <STG_INFO OSP_VER="10.2.0.0.0"/>
    <CMP_JAR_INFO>
        <INFO NAME="filemapObj" VAL="Components/oracle/server/v11_2_0_1_
0/filemap.xml"/>
        <INFO NAME="helpDir" VAL="Components/oracle/server/v11_2_0_1_0/help/"/>
        <INFO NAME="actionsClass" VAL="Components.oracle.server.v11_2_0_1_
0.CompActions"/>
        <INFO NAME="resourceClass" VAL="Components.oracle.server.v11_2_0_1_
0.resources.CompRes"/>
        <INFO NAME="identifiersXML" VAL="Components/oracle/server/v11_2_0_1_
0/identifiers.xml"/>
        <INFO NAME="contextClass" VAL="Components.oracle.server.v11_2_0_1_
0.CompContext"/>
        <INFO NAME="fastCopyLogXML" VAL="Components/oracle/server/v11_2_0_1_
0/fastCopyLog.xml"/>
    </CMP_JAR_INFO>
    <LOC_INFO INST_DFN_LOC="../Scripts" JAR_NAME="install2.jar"/>
    <BOOK NAME="oracle.server.hs"/>
    <PRE_REQ DEF="F"/>
    <PROD_HOME DEF="F"/>
    <DEP_GRP_LIST>
        <DEP_GRP NAME="Optional" EXT_NAME="Optional" TYPE="O">
            <DEP_LIST>
                <DEP NAME="oracle.rdbms" VER="11.2.0.1.0" PLAT="Linux"/>
                <DEP NAME="oracle.options" VER="11.2.0.1.0" PLAT="Linux"/>
```

```
                <DEP NAME="oracle.network" VER="11.2.0.1.0" PLAT="Linux"/>
                <DEP NAME="oracle.rdbms.oci" VER="11.2.0.1.0" PLAT="Linux"/>
                <DEP NAME="oracle.precomp" VER="11.2.0.1.0" PLAT="Linux"/>
                <DEP NAME="oracle.xdk" VER="11.2.0.1.0" PLAT="Linux"/>
                <DEP NAME="oracle.winprod" VER="11.2.0.1.0" PLAT="Linux"/>
                <DEP NAME="oracle.odbc" VER="11.2.0.1.0" PLAT="Linux"/>
                <DEP NAME="oracle.sysman.ccr" VER="10.2.4.0.0" PLAT="Linux"/>
            </DEP_LIST>
        </DEP_GRP>
        <DEP_GRP NAME="Required" EXT_NAME="Required" TYPE="R">
            <DEP_LIST>
                <DEP NAME="oracle.rdbms.install.common" VER="11.2.0.1.0"
PLAT="Linux"/>
                <DEP NAME="oracle.install.deinstalltool" VER="11.2.0.1.0"
PLAT="Linux"/>
            </DEP_LIST>
        </DEP_GRP>
        <DEP_GRP NAME="OptionalDecideNow" EXT_NAME="OptionalDecideNow" TYPE="O">
            <DEP_LIST>
                <DEP NAME="oracle.swd.oui" VER="10.1.0.2.0" PLAT="Linux"/>
                <DEP NAME="oracle.swd.opatch" VER="10.1.0.2.0" PLAT="Linux"/>
                <DEP NAME="oracle.dbjava.jdbc" VER="11.2.0.1.0" PLAT="Linux"/>
            </DEP_LIST>
        </DEP_GRP>
    </DEP_GRP_LIST>
    <DEP_LIST>
        <DEP NAME="oracle.rdbms" VER="11.2.0.1.0" DEP_GRP_NAME="Optional" HOME_
IDX="2"/>
        <DEP NAME="oracle.options" VER="11.2.0.1.0" DEP_GRP_NAME="Optional" HOME_
IDX="2"/>
        <DEP NAME="oracle.network" VER="11.2.0.1.0" DEP_GRP_NAME="Optional" HOME_
IDX="2"/>
        <DEP NAME="oracle.rdbms.oci" VER="11.2.0.1.0" DEP_GRP_NAME="Optional" HOME_
IDX="2"/>
        <DEP NAME="oracle.precomp" VER="11.2.0.1.0" DEP_GRP_NAME="Optional" HOME_
IDX="2"/>
        <DEP NAME="oracle.xdk" VER="11.2.0.1.0" DEP_GRP_NAME="Optional" HOME_
IDX="2"/>
        <DEP NAME="oracle.odbc" VER="11.2.0.1.0" DEP_GRP_NAME="Optional" HOME_
IDX="2"/>
        <DEP NAME="oracle.sysman.ccr" VER="10.3.1.1.0" DEP_GRP_NAME="Optional" HOME_
IDX="2"/>
        <DEP NAME="oracle.rdbms.install.common" VER="11.2.0.1.0" DEP_GRP_
NAME="Required" HOME_IDX="2"/>
        <DEP NAME="oracle.install.deinstalltool" VER="11.2.0.1.0" DEP_GRP_
NAME="Required" HOME_IDX="2"/>
        <DEP NAME="oracle.swd.oui" VER="11.2.0.1.0" DEP_GRP_NAME="OptionalDecideNow"
HOME_IDX="2"/>
        <DEP NAME="oracle.swd.opatch" VER="11.2.0.0.2" DEP_GRP_
NAME="OptionalDecideNow" HOME_IDX="2"/>
        <DEP NAME="oracle.dbjava.jdbc" VER="11.2.0.1.0" DEP_GRP_
NAME="OptionalDecideNow" HOME_IDX="2"/>
    </DEP_LIST>
    <REF_LIST>
        <REF NAME="oracle.server" VER="11.2.0.1.0" HOME_IDX="2"/>
    </REF_LIST>
    <INST_TYPE_LIST>
        <INST_TYPE NAME="EE" NAME_ID="EE" DESC_ID=""/>
    </INST_TYPE_LIST>
    <FILESIZEINFO>
```

```
                    <DEST VOLUME="%ORACLE_HOME%" SPACE_REQ="81045"/>
                    <DEST VOLUME="%INVENTORY_LOCATION%" SPACE_REQ="2000"/>
                    <DEST VOLUME="%s_TMPDIR_LOC%" SPACE_REQ="0"/>
                    <DEST VOLUME="%s_TEMP_LOC%" SPACE_REQ="0"/>
              </FILESIZEINFO>
          </COMP>
          </TL_LIST>
          <ONEOFF_LIST>
          </ONEOFF_LIST>
          <PATCHSET_UPDATE_LIST>
          </PATCHSET_UPDATE_LIST>
          </PRD_LIST>
```

E

Error Messages and Solutions

This appendix provides potential cause and possible solution information for all OUI errors. The information is arranged in numerical order according to the OUI error message numbers.

OUI - 10001

Message

```
OUI cannot determine the platform of this system. This may occur if OUI is running
on a system that is not supported or there is a bug in OUI. Please confirm that
the system is a supported configuration; if it is supported, please contact Oracle
Support.
```

Potential Causes

OUI determines the platform from the file
<ORACLE_HOME>/inventory/ContentsXML/oraclehomeproperties.xml

This message can occur from one of the following causes:

- OUI cannot detect the platform from the file
  <ORACLE_HOME>/inventory/ContentsXML/oraclehomeproperties.xml

- OUI is running on an unsupported system.

- There is an OUI bug.

Possible Solutions

- Check Metalink Certify to confirm that the system is a supported configuration.

- Contact Oracle Support.

OUI - 10004

Message

```
The target area <location_of_oraInventory> is being written to by another session.
Please close all the other(write) sessions for accessing inventory.
```

Potential Causes

For any write operations in the inventory, OUI acquires a lock. The file "writer.lock" is created under the <oraInventory>/locks directory. The error message appears if a lock exists in <oraInventory>/locks that another OUI has already acquired.

Possible Solutions

- Check if a "writer.lock" already exists in the <oraInventory>/locks directory.

- Check if another OUI is running in the system that can hold a lock on the oraInventory.

    For Unix, you can use this command:

    ```
    % ps -ef | grep runInstaller
    ```

    For Windows, you can use one of these commands:

    ```
    > tasklist | findstr /R "setup.exe"

    > tasklist | findstr /R "oui.exe"

    > tasklist | findstr /R "javaw.exe"
         (Make sure that this is the Java process related to OUI.)
    ```

- Stop the process, then try to start OUI again.

OUI - 10005

Message

```
You do not have sufficient permissions to access the inventory
<location_of_oraInventory>, unable to remove an existing inactive writer lock file
in the inventory. Make sure you have enough permission to the ''locks'' directory
to create/remove the writer.lock file.
```

Potential Causes

For any write operations in the inventory, OUI acquires a lock. The file "writer.lock" is created under the <oraInventory>/locks directory. The global inventory (oraInventory) was locked by another process and the writer.lock file still exists even after the process exited. The current OUI (process) is not able to remove the "writer.lock" file, because the permission for the current user does not allow this. Possibly, the current user is neither the owner nor the member of the group that owns the <oraInventory>/locks directory.

Possible Solutions

- Make sure you have write permission on the <oraInventory>/locks directory.

- Either change the ownership of <oraInventory>/ to the current user, or include the current user in the group that owns the <oraInventory> directory, provided that the group has write permission to the <oraInventory> directory.

OUI - 10020

Message

```
The target area <location_of_oraInventory> is being used as a source by another
session. A write lock cannot be obtained.
```

Potential Causes

For any write operations in the inventory, OUI acquires a lock. The file "writer.lock" is created under the <oraInventory>/locks directory. This error message appears if the <oraInventory> directory is accessed by another process (may be third party tools

also) so that OUI cannot create a "writer.lock" file in the <oraInventory>/locks directory.

Possible Solutions

Check if any process is accessing <oraInventory>. If so, kill/stop the process. For Unix, you can use this command:

```
/sbin/fuser <path_to_oraInventory>
```

OUI - 10021

Message

```
The target area <location_of_oraInventory> is being written to by another session.
A read lock cannot be obtained.
```

Potential Causes

Another process has locked the global inventory (oraInventory). For any reading operation or when OUI is started, OUI creates a "reader.lock" file in the <oraInventory>/locks directory. Since a write lock ("writer.lock") already exists in the <oraInventory>/locks directory, OUI cannot create a "reader.lock".

Possible Solutions

- Check if a "writer.lock" already exists in the <oraInventory>/locks directory.

- Check if another OUI is running in the system that can hold a lock on the oraInventory.

  For Unix, you can use this command:

  ```
  % ps -ef | grep runInstaller
  ```

  For Windows, you can use one of these commands:

  ```
  > tasklist | findstr /R "setup.exe"

  > tasklist | findstr /R "oui.exe"

  > tasklist | findstr /R "javaw.exe"
        (Make sure that this is the Java process related to OUI.)
  ```

- Stop the process, then try to start OUI again.

OUI - 10022

Message

```
The target area {0} cannot be used because it is in an invalid state.
```

Potential Causes

This error can occur for many reasons. The cause is usually global inventory (oraInventory) corruption.

Possible Solutions

- Make sure you have write permission on the <oraInventory> directory.

- Check the health of the inventory using this OPatch command:

```
% OPatch lsinventory

Verify that the command is able to list the details.
```

If the central inventory is corrupted, you can recreate it. For information about this, see Note 556834.1 - "Steps To Recreate Central Inventory (oraInventory) In RDBMS Homes" accessible through your My Oracle Support account.

OUI - 10024

Message

```
The DISPLAY environment variable is not set. The DISPLAY environment variable
should be set to <hostname or IP address>:<screen>, where the <screen> is usually
:0.0.
```

Potential Causes

The DISPLAY environment variable is not set, which OUI requires.

Possible Solutions

1. Check if the DISPLAY environment variable is set. Use the following command to view the current DISPLAY environment variable setting:

   ```
   echo $DISPLAY
   ```

2. If the DISPLAY environment variable is not set, use the following command to see what shell is being used:

   ```
   % echo $SHELL
   ```

3. Use one of the following commands for your Unix shell as an example to set the variable:

   For the csh shell:

   ```
   % setenv DISPLAY 192.168.1.128:0.0
   ```

   For sh, ksh, and bash shells:

   ```
   $ DISPLAY=192.168.1.128:0.0
   $ export DISPLAY
   ```

OUI - 10025

Message

```
Unable to start an interactive install session because of the following error:{0}
The DISPLAY environment variable should be set to <hostname or IP
address>:<screen>, where the <screen> is usually ''0.0".
```

Potential Causes

The DISPLAY environment variable is not set properly, or client users are not authorized to connect to the X Server.

Possible Solutions

1. Check if the DISPLAY environment variable is set. Use the following command to view the current DISPLAY environment variable setting:

    ```
    echo $DISPLAY
    ```

2. If the DISPLAY environment variable is not set, use the following command to see what shell is being used:

    ```
    % echo $SHELL
    ```

3. Use one of the following commands for your Unix shell as an example to set the variable:

    For the csh shell:

    ```
    % setenv DISPLAY 192.168.1.128:0.0
    ```

    For sh, ksh, and bash shells:

    ```
    $ DISPLAY=192.168.1.128:0.0
    $ export DISPLAY
    ```

4. Make sure that client users are authorized to connect to the X Server. To enable client users to access the X Server, open an xterm, dtterm, or xconsole as the user that started the session and type the following command:

    ```
    % xhost +
    ```

5. To test that the DISPLAY environment variable is set correctly, run an X11 based program that comes with the native operating system, such as 'xclock'.

    If you are unable to successfully run xclock, refer to your PC-X Server or OS vendor for further assistance.

OUI - 10028

Message

```
The directory location you have specified ( <location> ) is not registered as an
Oracle Home in the central inventory. Do you want to continue installation in this
Oracle Home?
```

Potential Causes

OUI is not able to retrieve the Oracle Home details from the Central inventory (oraInventory).

Possible Solutions

■   Check whether the Oracle home is present in the central inventory. Use the following command to determine this:

```
% OPatch lsinventory -all
```

■   If the Oracle home is not present in the central inventory, use one of the following applicable commands to register the Oracle home with the central inventory:

For Unix, use these commands:

```
% cd $ORACLE_HOME/oui/bin
%./runInstaller -attachHome ORACLE_HOME="<10g Oracle_Home Path>"
```

For Windows, use these commands:

```
> cd %ORACLE_HOME%/oui/bin
> setup.exe -attachHome ORACLE_HOME="<10g Oracle_Home Path>"
```

OUI - 10029

Message

You have specified a non-empty directory to install this product. It is recommended to specify either an empty or a non-existent directory. You may, however, choose to ignore this message if the directory contains Operating System generated files or subdirectories like 'lost+found'.

Potential Causes

The Oracle Home path (directory) specified for the installation already exists, and the directory contains files or subdirectories.

Possible Solutions

Select one of the following choices:

- Check if any directory/files exist inside the Oracle Home directory specified. Remove the directories/files if they are not required, then continue the installation. You can ignore directories like "lost+found".

- Specify another non-existent directory as the Oracle home. OUI creates it during installation.

- Specify a directory that is empty.

> **Note:** Use "-force" to continue past this error for a silent installation.

OUI - 10030

Message

You have specified a non-empty directory to install this product. It is recommended to specify either an empty or a non-existent directory. You may, however, choose to ignore this message if the directory contains Operating System generated files or subdirectories like 'lost+found'. Do you want to proceed with installation in this Oracle Home?

Potential Causes

The Oracle Home path (directory) specified for the installation already exists, and the directory contains files or subdirectories.

Possible Solutions

Select one of the following choices:

- Check if any directory/files exist inside the Oracle Home directory specified. Remove the directories/files if they are not required, then continue the installation. You can ignore directories like "lost+found".

- Specify another non-existent directory as the Oracle home. OUI creates it during installation.

- Specify a directory that is empty.

> **Note:** Use "-force" to continue past this error for a silent installation.

OUI - 10031

Message

Inventory location cannot be empty.

Potential Causes

This message appears only if you used the -invPtrLoc argument, as in runInstaller -invPtlLoc.

Possible Solutions

Specify the location for the central Inventory and continue with the installation.

OUI - 10033

Message

The inventory location <location> set by the previous installation session is no longer accessible. Do you still want to continue by creating a new inventory? Note that you may lose the products installed in the earlier session.

Potential Causes

There can be multiple causes for this error. Generally, most of the problems will be related to:

- Incorrect inventory location specified in the oraInst.loc file.

- Improper permissions on the inventory location specified in the oraInst.loc file.

Possible Solutions

Make sure that the inventory location specified in the oraInst.loc file is correct and also you have proper permissions on the specified location.

OUI - 10034

Message

Inventory location is invalid.

Potential Causes and Possible Solutions

There are several reasons this message can appear:

- The inventory location is truly invalid:

    - The oraInventory being pointed to through the inventory_loc reference in the /var/opt/oracle/oraInst.loc or /etc/oraInst.loc file or inst_loc in the registry is a non-existent location or has extra characters in the directory path listed.

      Fix — Search for the oraInventory directory on the system and change the location in oraInst.loc or inst_loc to point to the correct location, or if the path listed is incorrect, correct it.

- The oraInventory being pointed to through the inventory_loc reference in the /var/opt/oracle/oraInst.loc or /etc/oraInst.loc file or inst_loc in the registry is not the directory oraInventory.

  Fix — The directory pointed to needs to end with oraInventory (/usr/opt/oracle/oraInventory). Modify inventory_loc in oraInst.loc to point to the valid oraInventory directory or the inst_loc in the registry.

- The oraInventory being pointed to through the inventory_loc reference in the /var/opt/oracle/oraInst.loc or /etc/oraInst.loc file or inst_loc in the registry is oraInventory, but has no subdirectories.

  Fix — If the location has no subdirectories, it has been cleaned out accidentally. If you have an up-to-date backup of this area, try restoring the oraInventory; otherwise, you will need to reinstall all 8i, 9i, or 10g software versions that are on the host to repopulate the oraInventory.

■ The user attempting to perform the installation cannot access the inventory location.

The oraInventory being pointed to through the inventory_loc reference in the /var/opt/oracle/oraInst.loc or /etc/oraInst.loc file or inst_loc in the registry has permissions that restrict the user running the OUI from writing to oraInventory.

Fix — Ensure that you are the correct user to perform the installation. If so, then you will need to do a recursive chmod to enable access to the oraInventory:

```
$ chmod -R 770 <full path>/oraInventory
```

... and 660 for files under that.

Permissions might not be one of the causes, but the default permissions OUI creates are 770 for directories and 660 for files.

■ The group listed for group= in the /var/opt/oracle/oraInst.loc or /etc/oraInst.loc file is different than the group of the user running the OUI, and cannot access or write to oraInventory.

Fix — Ensure that you are in the correct group to perform the installation. If so, you will need to do a recursive chgrp to specify the correct group ownership of the oraInventory:

```
$ chgrp -R <correct group> <full path>/oraInventory
```

> **Note:** "inst_loc" on Windows should always point to %SYSTEMDRIVE%\Program Files\Oracle\Inventory. Manually changing this to point to another directory is unsupported.

OUI - 10035

Message
```
You do not have permission to write to the inventory location.
```

Potential Causes
The inventory location is not accessible by the user attempting to perform the installation. The oraInventory being pointed to through the inventory_loc reference in the /var/opt/oracle/oraInst.loc or /etc/oraInst.loc file or inst_loc in registry has permissions that restrict the user running the OUI from writing to the oraInventory.

Possible Solutions

Ensure that you are the correct user to perform the installation. If so, you will need to do a recursive chmod to enable access to the oraInventory:

```
$ chmod -R 770 <full path>/oraInventory
```

OUI - 10036

Message

```
Could not create the inventory location. You may not have permission to write to
this location.
```

Potential Causes

The Oracle user does not have proper permission for the inventory location specified during installation.

Possible Solutions

Ensure that you are the correct user to perform the installation, and that you have proper permissions in the specified directory. Check this using the following command:

```
% ls -l <location>
```

If you do not have proper permissions, provide them as shown:

```
% chmod -R 755 <full path>
```

OUI - 10038

Message

```
You do not have the necessary permissions to write to the inventory at <LOCATION>.
Please make sure that you have the appropriate permissions to perform the
installation.
```

Potential Causes

You do not have write permission for the inventory location specified during installation.

Possible Solutions

Ensure that you are the correct user to perform the installation, and that you have proper permissions in the specified directory. Check this using the following command:

```
% ls -l <location>
```

If you do not have proper permissions, provide them as shown:

```
% chmod -R 755 <full path>
```

OUI - 10039

Message

```
Unable to read install inventory <Location> on this system. Please check if you
```

have the proper permissions to read/write to the inventory.

Potential Causes

You do not have read permission for the inventory location specified during installation.

Possible Solutions

Ensure that you are the correct user to perform the installation, and that you have proper permissions in the specified directory. Check this using the following command:

```
% ls -l <location>
```

If you do not have proper permissions, provide them as shown:

```
% chmod -R 755 <full path>
```

OUI - 10040

Message

```
Unable to read inventory pointer file: Check if you have the proper permissions.
```

Potential Causes

The inventory pointer (oraInst.loc) is owned by "root" and other users have read permission for it. For example:

```
[oracle@oracle ~]$ ls -l /etc/oraInst.loc
-rw-r--r-- 1 root root 65 Oct 3 18:42 /etc/oraInst.loc
```

But in this case, there is no read permission for the Oracle user on the inventory pointer.

Possible Solutions

Ensure that you are the correct user to perform the inventory actions, and that you have proper permissions in the inventory pointer (oraInst.loc). Check this using the following command:

```
% ls -l <location>
```

If you do not have proper permissions, provide them as shown:

```
% chmod -R 644 <location>
```

OUI - 10041

Message

```
Unable to read inventory location registry entry. Check if the registry entry
exists and if you have the proper permissions.
```

Potential Causes

Accessing the inventory pointer (orainst_loc) key requires administrator privileges; other users will have read permission.

Possible Solutions

The following command displays the central inventory pointer (orainst_loc) key and value:

```
C:\>reg query HKLM\SOFTWARE\ORACLE /v inst_loc
```

Make sure that you have administrator privileges to read the registry HKLM\SOFTWARE\ORACLE\inst_loc.

OUI - 10044

Message

```
The selected location already contains an Oracle Home or APPL_TOP created while
running a different OS.
```

Potential Causes

This error can occur for the following reasons:

- The software/patchset you are trying to install is intended for a different platform.

- The file <Central_inventory>/install.platform is corrupted.

Possible Solutions

Make sure that the software/patchset you are trying install is intended for the same platform, and make sure that the <Central_inventory>/install.platform file is not corrupted.

The install.platform file is written to only during inventory creation and does not become corrupted unless someone manually changes it.

OUI - 10053

Message

```
Unable to generate temporary script: Unable to continue install.
```

Potential Causes

This error usually occurs when generating the orainstRoot.sh script in the "/tmp" directory. You may not have proper permissions in the /tmp directory, or /tmp is full.

Possible Solutions

- Check the permissions of the /tmp directory:

  ```
  % ls -l /tmp
  ```

- Make sure that the /tmp directory has the proper write permissions. Run this command:

  ```
  % chmod 0777 /tmp ( as root )
  ```

- Check if the /tmp directory has enough space for creating the file.

OUI - 10054

Message

```
Unable to modify the group ownership of the OUI inventory to the requested group
name. Either the specified group doesn't exist, or the current userid does not
belong to that group. If the group exists, you may run the shell script as root to
change the group name after installation is complete.
```

Potential Causes

This error usually occurs when the root.sh script attempts to modify the Central inventory group ownership.

Possible Solutions

- Make sure that the group specified at the start of the installation is an existing group.

- Make sure that you are a part of the particular group specified at the start of installation. Check the group "dba" as shown in the following example:

```
%cat /etc/group | grep dba
dba:x:502:oracle
```

In this example, the group is dba, the group ID is 502, and the user oracle is a part of the dba group.

OUI - 10055

Message

```
Error in accessing the registry.
```

Potential Causes

This error usually occurs when you do not have administrator rights/permissions on the registry.

Possible Solutions

- Make sure that you have administrator privileges to access/modify the registry.

- Check the permissions in regedit —> hkey_local_machine —> software —> oracle (Permissions - advanced tab).

OUI - 10056

Message

```
Error in updating the registry.
```

Potential Causes

This error usually occurs when you do not have administrator rights/permissions on the registry.

Possible Solutions

- Make sure that you have administrator privileges to access/modify the registry.

■ Check the permissions in regedit —> hkey_local_machine —> software —> oracle (Permissions - advanced tab).

OUI - 10058

Message

```
The OUI Inventory on this system does not exist or is not writable. The possible
POTENTIAL/POSSIBLE CAUSES are:
1. The script <path>/orainstRoot.sh was not executed with root privileges, or an
error occurred during execution.
2. You are not a member of the specified installer group.
```

Potential Causes

One of the following situations may apply:

■ The permissions for the orainstRoot.sh script are not set correctly for execution as "root" user.

■ You are not part of the particular group specified at the start of installation.

Possible Solutions

■ Make sure that the script is run as user "root."

■ Make sure that you are a part of the particular group specified at the start of installation. Check the group "dba" as shown in the following example:

```
%cat /etc/group | grep dba
dba:x:502:oracle
```

In this example, the group is dba, the group ID is 502, and the user oracle is a part of the dba group.

OUI - 10059

Message

```
English cannot be removed from the selected languages. English files are always
installed.
```

Potential Causes

This message appear when you try to deselect English. If multiple languages are defined for the installed products, Oracle Universal Installer enables language selection for all installation types through the "Languages" button.

If you click on the Product Languages, the screen with a selection/deselection list appears. In this situation, English is selected by default and you cannot deselect it, because English files are always installed.

Possible Solutions

Do not attempt to deselect English from the language list, as English files are always installed.

OUI - 10070

Message

```
The patch, which is part of patch set, has not been selected for installation.
```

Potential Causes

You may be trying to install the binaries on top of an incorrect Oracle home, such as installing 11.1.0.7.0 on top of 10.2.0.4.0.

Possible Solutions

Install the base software (11.1.0.6.0 in this example) in a separate Oracle home and the upgrade the database.

OUI - 10082

Message

```
The staging area needs Oracle Universal Installer version <version> or later.
```

Potential Causes

The OUI you are trying to use is an older version, so the installation cannot continue. For instance, you may encounter this error when trying to install Oracle RDBMS 11.2 or later with an 11.1.0.6 OUI.

Possible Solutions

Use the OUI version specific to the version of the Oracle software. For instance, installation of Oracle RDBMS 11.2 requires OUI 11.2 or later.

OUI - 10083

Message

```
Some of the components you selected cannot be deinstalled because they were
installed using Oracle Universal Installer version {0}. Please use Oracle
Universal Installer version {0} or higher to deinstall these products.
```

Potential Causes

The OUI you are trying to use is an older version, so the de-installation cannot continue. For instance, you may encounter this error when trying to de-install Oracle RDBMS 10.2 or later with a 10.1 OUI.

Possible Solutions

Use the OUI version specific to the version of the Oracle software for de-installation. For example, de-installation of Oracle RDBMS 11.2 requires OUI 11.2 or later.

OUI - 10088

Message

```
This is a required component. You cannot de-select a required component.
```

Potential Causes

This error can occur when you attempt to de-select a required component. There may be dependencies with the component you are trying to de-select. For example, this message appears when you are doing a custom installation of Oracle RDBMS and are trying to de-select a required component that is required for the functioning of the Oracle RDBMS kernel.

Possible Solutions

It is not possible to de-select a required component.

OUI - 10091

Message

```
There are no patches that need to be applied from the patchset Oracle <version>
Patchset "x.x.x.x.x."
```

Potential Causes

- Oracle Software might already be installed on x.x.x.x.x.

- You might be applying the patchset x.x.x.x.x on an incorrect Oracle home.

- The local inventory may be corrupted.

Possible Solutions

- Start SQL*Plus and check the banner for the version to make sure that the version of the software is correct.

- Make sure that you are applying the patchset on top of the correct Oracle home.

- Make sure that the inventory is healthy. Check if OPatch lists all of the products:

  ```
  % OPatch lsinventory -detail
  ```

- If the local inventory is corrupted, you need to reinstall the software, then continue with the installation.

OUI - 10092

Message

```
The selected patches cannot be applied in this Oracle Home. Please try installing
them in a different Oracle Home.
```

Potential Causes

- The patch/patchset you are trying to install is not suited for the particular Oracle home.

- The patchset may be a different version; for instance, when you are trying to install a 11.1.0.7.0 Oracle home on top of a 10.2.0.4.0 Oracle home.

- The patchset you are trying to install is a lower version, such as installing an 11.1.0.7.0 patchset on top of an 11.2 Oracle home.

Possible Solutions

- Check the version of the Oracle home:

```
% <ORACLE_HOME>/OPatch/OPatch lsinventory -detail
```

■ Make sure that the patchset is suited for the version of the Oracle home.

OUI - 10094

Message

```
Problem in saving inventory. Installation cannot continue.
```

Potential Causes

This error message can occur because you do not have proper permissions on the inventory, or there is an OUI bug.

Possible Solutions

Before you install the 11.2 software, make sure you have sufficient write permissions on the central inventory and the local inventories of all existing ORACLE_HOMEs listed in the central inventory.

OUI - 10095

Message

```
You may not have enough space to write to the inventory. Do you still want to
continue?
```

Potential Causes

This error occurs when the space is full in the disk where the inventory resides.

Possible Solutions

Make sure that you have sufficient space in the disk before you continue with the installation. Otherwise, continuing may lead to inventory corruption.

OUI - 10096

Message

```
You may not have enough space to write to <location>. Please create more space at
this location and try again.
```

Potential Causes

The disk is full where the Oracle home is being installed or already installed. The base software/patchset installation can therefore not continue.

Possible Solutions

Add more space to the disk and try the installation again.

OUI - 10104

Message

```
Some of the configuration assistants failed. It is strongly recommended that you
retry the configuration assistants at this time. Not successfully running any
```

recommended assistants means your system will not be correctly configured. Select
the failed assistants and click the 'Retry' button to retry them.

Potential Causes

Some of the recommended Configuration Assistants failed during installation.

Possible Solutions

Retry the failed Configuration Assistants.

OUI - 10105

Message

Some of the configuration assistants failed. However, these are optional
assistants, so they are not required for the correct configuration of your system.
If you want to try to run those assistants again, select the failed assistants and
click the 'Retry' button.

Potential Causes

Some of the optional Configuration Assistants failed during installation.

Possible Solutions

Retry the failed Configuration Assistants or invoke the Configuration Assistant after
the installation. For example, for DBCA:

<ORACLE_HOME>/bin/dbca

OUI - 10116

Message

Unable to synchronize home. Not enough permission to write to directory.

Potential Causes

OUI attempts to synchronize all the Oracle homes at the end of the installation. This
message can occur when the user installing Oracle does not have sufficient
permissions on the other Oracle home.

Possible Solutions

Make sure that you have sufficient permissions for the Oracle homes, or you are from
the group specific to all the installations.

OUI - 10122

Message

There was a problem in accessing the staging area. File not found.

Potential Causes

OUI expects certain files to be present in the staging area, but OUI could not find them
because the staging area may be corrupted.

Possible Solutions

Make sure that the downloaded software (staging area) is not corrupted.

OUI - 10135

Message

```
This inventory requires Oracle Universal Installer version {0} or higher.
```

Potential Causes

The OUI you are trying to use is an older version, so the inventory operations cannot continue. For example, you may encounter this error when trying to update the inventory of Oracle RDBMS 11.2 or later with an 11.1.0.6 OUI.

Possible Solutions

Use the OUI specific to the version of the Oracle software for de-installation.

OUI - 10136

Message

```
An <OH_Name> with name <NAME> already exists at location. Please specify another
location for <path>.
```

Potential Causes

An Oracle home already exists in the path selected for the installation.

Possible Solutions

Select another path for the installation, or de-install the Oracle home already installed in the particular path (if no longer required) and perform the new installation.

OUI - 10137

Message

```
An Oracle home with name <Name> already exists at location. Please specify another
name.
```

Potential Causes

An Oracle home with the specified name already exists in the inventory.

Possible Solutions

Specify another name for the Oracle home and continue with the installation.

OUI - 10153

Message

```
The Oracle home is not valid for cloning. Install-time information cannot be
retrieved for component.
```

Potential Causes

This error can occur because some of the components are not properly installed or the inventory is not properly updated.

Possible Solutions

Either correctly install the missing components in the source ORACLE_HOME or remove them prior to cloning.

Index

V

W

X