Oracle® Real Application Clusters

Installation Guide

12*c* Release 1 (12.1) for Linux and UNIX

E48917-11

August 2015

ORACLE®

Oracle Real Application Clusters Installation Guide, 12*c* Release 1 (12.1) for Linux and UNIX

E48917-11

Copyright © 2013, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Aparna Kamath

Contributing Authors: Douglas Williams, Mark Bauer, Jonathan Creighton, John McHugh, Markus Michalewicz, Satish Panchumarthy, Janet Stern

Contributors: David Austin, Mark Bauer, Mark Fuller, Donald Graves, Prakash Jashnani, Kevin Jernigan, Aneesh Khandelwal, Bryn Llewellyn, Mughees Minhas, Srinivas Poovala, Sampath Ravindhran, Janelle Simmons, James Spiller, Malai Stalin, Richard Strohm, Roy Swonger, Siu Tam, Ara Shakian, Jingci Wang, James Williams, Ying Zhang, Michael Zoll,Rajesh Prasad,Kannan Viswanathan

Contributor: The Database 12c documentation is dedicated to Mark Townsend, who was an inspiration to all who worked on this release.

Contents

4   Deploying Oracle RAC Databases with Rapid Home Provisioning

5   Oracle Real Application Clusters Postinstallation Procedures

6 Using Server Pools with Oracle RAC

7 Understanding the Oracle RAC Installed Configuration

8   Removing Oracle Real Application Clusters Software

A   Troubleshooting Oracle Real Application Clusters Installations

B   Using Scripts or Response Files to Create Oracle RAC Databases

C   Directory Structure for Oracle RAC Environments

D   How to Stop Processes in an Existing Oracle RAC Database

E   Managing Oracle Database Port Numbers

Index

List of Tables

Preface

Oracle Real Application Clusters Installation Guide for Linux and UNIX explains how to install and configure Oracle Real Application Clusters (Oracle RAC). Before you use this guide, you must first complete an installation of Oracle Clusterware, as described in the *Oracle Grid Infrastructure Installation Guide* for your platform.

Intended Audience

Oracle Real Application Clusters Installation Guide for Linux and UNIX provides database installation information for database administrators (DBAs) who install and configure Oracle RAC.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, refer to the following Oracle resources:

Oracle Real Application Clusters Documentation

Most Oracle error message documentation is only available in HTML format. If you only have access to the Oracle Documentation media, then browse the error messages by range. Once you find a range, use your browser's "find in page" feature to locate a specific message. When connected to the Internet, you can search for a specific error message using the error message search feature of the Oracle online documentation.

Installation Guides

- *Oracle Diagnostics Pack Installation Guide*
- *Oracle Grid Infrastructure Installation Guide* for your platform

Operating System-Specific Administrative Guides

- *Oracle Clusterware Administration and Deployment Guide*

- *Oracle Database Administrator's Reference, 12c Release 1 (12.1) for UNIX Systems*

- *Oracle Database Platform Guide for Microsoft Windows*

Oracle Database 12*c* Release 1 (12.1) Real Application Clusters Management

- *Oracle Real Application Clusters Administration and Deployment Guide*

- *Oracle Database 2 Day DBA*

- *Oracle Enterprise Manager Real Application Clusters Guide Online Help*

- *Getting Started with the Oracle Diagnostics Pack*

Generic Documentation

- *Oracle Database New Features Guide*

- *Oracle Database Net Services Administrator's Guide*

- *Oracle Database Concepts*

- *Oracle Database Reference*

Printed documentation is available for sale in the Oracle Store at the following website:

https://shop.oracle.com

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at the following website:

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

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN website:

http://www.oracle.com/technetwork/indexes/documentation/index.html

Oracle error message documentation is available only in HTML. You can browse the error messages by range in the Documentation directory of the installation media. When you find a range, use your browser's search feature to locate a specific message. When connected to the Internet, you can search for a specific error message using the error message search feature of the Oracle online documentation.

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.

Changes in This Release for Oracle RAC

This chapter lists changes in *Oracle Real Application Clusters Installation Guide*.

Changes in Oracle Database 12*c* Release 1

The following are changes in *Oracle Real Application Clusters Installation Guide* for Oracle Database 12*c* Release 1.

- New Features for Oracle Database 12c Release 1 (12.1.0.2)
- New Features for Oracle Database 12c Release 1 (12.1.0.1)

New Features for Oracle Database 12c Release 1 (12.1.0.2)

- Standard Edition Replacement

  Starting with Oracle Database 12*c* Release 1(12.1.0.2), Standard Edition 2 (SE2) replaces Standard Edition. Standard Edition 2 can run on systems with up to two sockets and can support a two-node Oracle RAC cluster.

  > **See Also:** My Oracle Support Note 2027072.1 for more information:
  >
  > https://support.oracle.com/epmos/faces/DocumentDisplay?id=20 27072.1&displayIndex=1

- Oracle ASM Filter Driver

  The Oracle ASM filter driver (Oracle ASMFD) is a kernel module that resides in the I/O path of the Oracle ASM disks. It is used by Oracle ASM to specify the parameters to validate all I/O requests.

  The filter driver rejects any I/O requests that are not determined to be valid. This action eliminates accidental overwrites of Oracle ASM disks that would cause corruption in an Oracle ASM disk and files within the disk group. For example, the driver filters out all non-Oracle I/O requests that could cause accidental overwrites.

  See *Oracle Automatic Storage Management Administrator's Guide* for more information about configuration and administration of Oracle ASMFD.

- Rapid Home Provisioning

  Rapid Home Provisioning is a method of deploying software homes to nodes in a cloud computing environment from a single cluster where you store home images (called gold images) of Oracle software, such as databases, middleware, and applications. Rapid Home Provisioning Server (RHPS) clusters provide gold images to Rapid Home Provisioning Clients (RHPC).

> **See Also:** ▪ Chapter 4, "Deploying Oracle RAC Databases with Rapid Home Provisioning"
>
> ▪ *Oracle Clusterware Administration and Deployment Guide*

- Cluster and Oracle RAC Diagnosability Tools Enhancements

  The Trace File Analyzer (TFA) Collector is installed automatically with Oracle Grid Infrastructure installation. The Trace File Analyzer Collector is a diagnostic collection utility to simplify diagnostic data collection on Oracle Grid Infrastructure and Oracle RAC systems.

  > **See Also:** *Oracle Clusterware Administration and Deployment Guide* for information about using Trace File Analyzer Collector

- Automatic Installation of Grid Infrastructure Management Repository

  The Grid Infrastructure Management Repository is automatically installed with Oracle Grid Infrastructure 12c Release 1 (12.1.0.2).

- Oracle RAC Cache Fusion Accelerator

  Oracle RAC uses its Cache Fusion protocol and Global Cache Service (GCS) to provide fast, reliable, and efficient inter-instance data communication in an Oracle RAC cluster, so that the individual memory buffer caches of multiple instances can function as one global cache for the database. Using Cache Fusion provides a nearly linear scalability for most applications. This release includes accelerations to the Cache Fusion protocol that provide enhanced scalability for all applications.

New Features for Oracle Database 12c Release 1 (12.1.0.1)

- Database Upgrade Automation Using DBUA

  There are three areas that are being enhanced for upgrade ease-of-use. First, in the pre-upgrade phase, the existing manual steps are eliminated and give more explicit advice or even generate a fix-up script to find issues identified in the pre-upgrade phase. Second, in the post-upgrade phase, there is a post-upgrade health check that indicates that the upgrade was successful. Finally, partner documents (such as SAP) and major customer upgrade documents are used to further identify manual steps that may be automated and generalized to a wider customer base.

  Automating the upgrade process provides major improvements in usability and ease-of-use. There is also better integration of database upgrade with Oracle Grid Infrastructure for a cluster and Oracle Enterprise Manager Cloud Control.

  See *Oracle Database Upgrade Guide*.

- DBCA Support for Multitenant Container Database and Pluggable Database Configurations

  Starting with Oracle Database 12c Release 1 (12.1), Oracle Database Configuration Assistant (DBCA) allows you to create either a multitenant container database (CDB) or a non-CDB. You can create the CDB with zero, one, or more pluggable databases (PDBs).

  You can also create a CDB with one PDB during the database installation.

  See *Oracle Database Administrator's Guide*.

- Enhancements to Cluster Health Monitor (CHM)

CHM has been enhanced to be more efficient to support Oracle Flex Clusters implementations. These enhancements ensure that Oracle Flex Clusters run smoothly while minimizing the required resources to monitor the stack.

See *Oracle Clusterware Administration and Deployment Guide*.

- Oracle Flex ASM Servers

  Oracle Flex ASM enables the Oracle ASM instance to run on a separate physical server from the database servers. Many Oracle ASM instances can be clustered to support a large number of database clients.

  Note that Oracle Flex ASM can apply to a collection of databases, each one a single instance but running in a Flex ASM Cluster.

  See *Oracle Automatic Storage Management Administrator's Guide*.

- Oracle Flex Clusters

  Oracle Flex Cluster is a new concept, which joins together a traditional closely coupled cluster with a modest node count with a large number of loosely coupled nodes. In order to support various configurations that can be established using this new concept, SRVCTL provides new commands and command options to ease the installation and configuration.

  See *Oracle Clusterware Administration and Deployment Guide*.

- IPv6 Support for Public Networks

  Oracle Clusterware 12*c* Release 1 (12.1) supports IPv6-based public IP and VIP addresses.

  IPv6-based IP addresses have become the latest standard for the information technology infrastructure in today's data centers. With this release, Oracle RAC and Oracle Grid Infrastructure support this standard. You can configure cluster nodes during installation with either IPv4 or IPv6 addresses on the same network. Database clients can connect to either IPv4 or IPv6 addresses. The Single Client Access Name (SCAN) listener automatically redirects client connects to the appropriate database listener for the IP protocol of the client request.

  See *Oracle Grid Infrastructure Installation Guide*.

- Multiprocess Multithreaded Oracle Database

  Starting with Oracle Database 12*c*, Oracle Database may use operating system threads to allow resource sharing and reduce resource consumption.

  See *Oracle Database Concepts*.

- Oracle ACFS Auditing and Support for Importing Auditing Data into Oracle Audit Vault Server

  This feature provides auditing for Oracle ACFS security and encryption. In addition, this feature also generates an XML file containing Oracle ACFS audit trail data which can be imported by Oracle Audit Vault Server.

  See *Oracle Automatic Storage Management Administrator's Guide*.

- Oracle Enterprise Manager Database Express 12c

  Oracle Database 12*c* introduces Oracle Enterprise Manager Database Express, a web management product built into Oracle Database without any need for special installation or management. Using Oracle Enterprise Manager Database Express, you can perform administrative tasks such as managing user security, and

managing database memory and storage. You can also view performance and status information about your database.

Note that starting with Oracle Database 12*c*, Oracle Enterprise Manager Database Control is deprecated.

See *Oracle Database 2 Day DBA*.

■   Policy-Based Cluster Management and Administration

Oracle Grid Infrastructure allows running multiple applications in one cluster. Using a policy-based approach, the workload introduced by these applications can be allocated across the cluster using a policy. In addition, a policy set enables different policies to be applied to the cluster over time as required. Policy sets can be defined using a web-based interface or a command-line interface.

Hosting various workloads in the same cluster helps to consolidate the workloads into a shared infrastructure that provides high availability and scalability. Using a centralized policy-based approach allows for dynamic resource reallocation and prioritization as the demand changes.

See *Oracle Clusterware Administration and Deployment Guide*.

■   Simplified Oracle Database Vault Installation

Starting with Oracle Database 12*c*, Oracle Database Vault is installed by default as part of the Oracle Database installation. However, you can configure, enable, or disable Oracle Database Vault after the Oracle Database installation, either using DBCA, or by running SQL statements.

See *Oracle Database Vault Administrator's Guide*.

■   Support for Separation of Database Administration Duties

Oracle Database 12*c* provides support for separation of administrative duties for Oracle Database by introducing task-specific and least-privileged administrative privileges that do not require the SYSDBA administrative privilege. These new privileges are: SYSBACKUP for backup and recovery, SYSDG for Oracle Data Guard, and SYSKM for encryption key management.

See *Oracle Database Security Guide*.

■   Unified Database Audit Configuration

Starting with Oracle Database 12*c*, you can create named audit policies. An audit policy contains a set of audit options, which is stored in the database as an object. The advantage of creating a named audit policy is that it reduces the number of commands that are required to create a database audit policy, and it simplifies the implementation of an audit configuration for security and compliance with conditional auditing. This new audit policy framework is included with the database installation.

See *Oracle Database Security Guide*.

Deprecated Features

The following features are deprecated in this release, and may be desupported in another release. See *Oracle Database Upgrade Guide* for a complete list of deprecated features in this release.

■   Deprecation Announcement for Oracle Restart

Oracle Restart is a feature provided as part of Oracle Grid Infrastructure. Oracle Restart monitors and can restart Oracle Database instances, Oracle Net Listeners,

and Oracle ASM instances. Oracle Restart is currently restricted to manage single instance Oracle Databases and Oracle ASM instances only, and is subject to desupport in future releases. Oracle continues to provide Oracle ASM as part of the Oracle Grid Infrastructure installation for Standalone and Cluster deployments.

- Deprecation of single-letter SRVCTL command-line interface (CLI) options

  All SRVCTL commands have been enhanced to accept full-word options instead of the single-letter options. All new SRVCTL command options added in this release support full-word options, only, and do not have single-letter equivalents. The use of single-letter options with SRVCTL commands might be desupported in a future release.

- Change for Standalone Deinstallation Tool

  The deinstallation tool is now integrated with the database installation media.

  See Section 8.3.1, "About the Deinstallation Tool"

- Deprecation of -cleanupOBase

  The -cleanupOBase flag of the deinstallation tool is deprecated in this release. There is no replacement for this flag.

- Oracle Enterprise Manager Database Control is replaced by Oracle Enterprise Manager Database Express.

- The deinstall standalone utility is replaced with a deinstall option using Oracle Universal Installer (OUI).

Desupported Features

The following features are no longer supported by Oracle. See *Oracle Database Upgrade Guide* for a complete list of features desupported in this release.

- Oracle Enterprise Manager Database Control

- CLEANUP_ORACLE_BASE property removed and does not support an Oracle base removal during silent or response file mode deinstalls.

Other Changes

The following are additional changes in the release:

- Oracle Grid Infrastructure and Oracle Real Application Clusters are not supported for 32-bit hardware platforms.

- PMON (process monitor) has been replaced with LREG (listener registration process) to notify listeners about instances, services, handlers and endpoints.

1

Preinstallation Checklist

This chapter is intended for database administrators to use in consultation with system and storage administrators to coordinate the installation and plan the tasks for Oracle Clusterware, in preparation for completing an installation of Oracle Real Application Clusters (Oracle RAC). This chapter contains the following topics:

- Before You Start: Planning Your Installation
- Overview of Server Hardware, Network, and Operating System
- Overview of Oracle RAC Storage
- Additional Information to Prepare for Installation
- Confirming Cluster Readiness for Oracle RAC Installation with CVU
- Installing Oracle Database 12c on Systems with Existing Databases

1.1 Before You Start: Planning Your Installation

This section provides a list of tasks that Oracle recommends you complete before starting Oracle Clusterware and Oracle RAC installation. Whether your location is a data center with a large project team of system administrators, storage administrators, network administrators, database administrators, and third-party hardware and software vendors, or you are a project team of one, planning is important to help ensure that your installation proceeds smoothly.

It is beyond the scope of this documentation set to advise how to determine hardware sizing or capacity planning for your installation. Note that with Oracle Clusterware and Oracle RAC, you can add nodes and instances as needed in response to testing, or in response to increased workloads.

Review and complete the following steps as part of your installation plan:

- Check My Oracle Support
- Check Oracle Technology Network
- Decide The Deployment Method for Oracle RAC and Oracle RAC One Node
- Review Installation Steps for Oracle RAC
- Review Documentation
- Select a Security Notification Contact
- Review Root Task Delegation Options With Your System Administrator
- Review Installation Fixup Scripts With Your System Administrator
- Review How to Use Cluster Verification Utility (CVU)

- Review Existing Oracle Installations and Upgrade Plans

- Consider Memory Allocation and Automatic Memory Management

- Review Globalization Requirements

1.1.1 Check My Oracle Support

Before you decide whether to install Oracle Database 12*c* Release 1 (12.1), log on to My Oracle Support:

https://support.oracle.com

You must register online before using My Oracle Support.

> **Note:** Contact your Oracle Sales Representative if you do not have a My Oracle Support account.

To access certifications for your installation:

1. Log on to My Oracle Support.

2. Click the **Certifications** tab. If you do not see a tab titled Certifications, then click **More** and select **Certifications** from the list.

3. In the **Certification Search** frame, select Oracle Real Application Clusters, select the release number, and select your operating system platform on which you intend to install. When you have completed your selections, click **Search** to generate a product certification list for your installation.

1.1.2 Check Oracle Technology Network

The Oracle Technology Network (OTN) contains white papers about deployment options, capacity planning, best practices on various NFS platforms, and extended clusters deployments, which are not addressed in this guide. You can review available papers at the following website:

http://www.oracle.com/goto/rac

In particular, check the Oracle RAC Technologies Certification Matrix for your platform:

Linux:

http://www.oracle.com/technetwork/database/clustering/tech-generic-linux-new-086754.html

UNIX:

http://www.oracle.com/technetwork/database/clustering/tech-generic-unix-new-166583.html

1.1.3 Decide The Deployment Method for Oracle RAC and Oracle RAC One Node

You can choose from the following deployment methods to install Oracle Real Applications (Oracle RAC) and Oracle RAC One Node software and provision databases:

❑ Deploy Oracle RAC or Oracle RAC One Node software in any of the following ways:

- Install Oracle RAC or Oracle RAC One Node Database software using Oracle Universal Installer (OUI). See Chapter 2, "Installing Oracle RAC and Oracle RAC One Node" for the procedure to install Oracle RAC using OUI.

- Provision Oracle RAC Or RAC One Node Database software using Rapid Home Provisioning. See Chapter 4, "Deploying Oracle RAC Databases with Rapid Home Provisioning" for more information about Rapid Home Provisioning.

❏ Deploy Oracle Database software and create Oracle RAC or Oracle RAC One Node databases in any of the following ways:

- Install Oracle RAC or Oracle RAC One Node Database using Oracle Universal Installer (OUI). See Chapter 2, "Installing Oracle RAC and Oracle RAC One Node" for the procedure to install Oracle RAC using OUI.

- Provision Oracle RAC Or RAC One Node Database using Rapid Home Provisioning. See Chapter 4, "Deploying Oracle RAC Databases with Rapid Home Provisioning" for more information about Rapid Home Provisioning.

❏ Create Oracle RAC database in an already-installed Oracle home using Database Configuration Assistant (DBCA). See Chapter 3, "Creating Oracle RAC or Oracle RAC One Node Databases with DBCA" for more information about creating database using DBCA.

1.1.4 Review Installation Steps for Oracle RAC

Installing Oracle RAC consists of the following steps:

1. Prepare servers (system, users and groups, network, and storage administration), as described in *Oracle Grid Infrastructure Installation Guide* for your platform. These tasks include the following:

   - Install the operating system and install the operating system packages and patches to the required version.

   - Create the required groups, users, and software homes.

   - Set up the domain name forwarding for Grid Naming Service (GNS) if you plan to deploy GNS or Multi-Cluster GNS, and set up the network addresses in the DNS and on the server as needed, depending on the cluster node type (Standard, Hub, Leaf) that you plan to configure for each node.

   - Set up the required storage.

   - (Optional) Stage all of the software on one node for installation (the "local node").

> **Note:** During Oracle Grid Infrastructure installation, you can select multiple interconnects for private network use. This feature is called Redundant Interconnect Usage.
>
> With Redundant Interconnect Usage, Oracle Clusterware can define multiple interfaces to use for the cluster interconnect network, without the need of using bonding or other technologies.
>
> When you define multiple interconnects, either during the installation interview or afterward using the Oracle Interface Configuration (OIFCFG) command line utility, Oracle Clusterware creates from one to four highly available IP (HAIP) addresses. Oracle RAC and Oracle Automatic Storage Management (Oracle ASM) instances use these interconnect addresses to ensure highly available, load-balanced interconnect communication between nodes.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for more information about using OIFCFG to modify interconnects, and *Oracle Grid Infrastructure Installation Guide* for your platform for more information about network configuration requirements

2. Install Oracle Grid Infrastructure for a cluster, which includes Oracle Clusterware and Oracle ASM (system and storage administration):

During installation, Fixup scripts perform additional configuration of operating system parameters, secure shell (SSH) for installation and user environment variables.

- To configure Oracle RAC on an Oracle Flex Cluster, Oracle Grid Infrastructure for a cluster must be installed and configured in Oracle Flex Clusters mode.

> **Note:**
>
> - You must have Oracle Grid Infrastructure installed successfully before you attempt to install Oracle RAC.
>
> - For Oracle Flex Clusters deployments, you cannot install Oracle RAC binaries on Oracle Automatic Storage Management Cluster File System (Oracle ACFS).

3. Install Oracle RAC (database administration):

- Install Oracle RAC or Oracle RAC One Node.

- Complete the postinstallation configuration of the Oracle RAC database.

1.1.5 Review Documentation

Before installing Oracle RAC, review *Oracle Grid Infrastructure Installation Guide* for your operating system platform to confirm that all cluster member nodes are configured with required users and groups, and that the system, storage, and network administrators have completed any other tasks required for an Oracle RAC installation.

In addition, review the Release Notes and My Oracle Support (https://support.oracle.com) to ensure that you have the most current information

about system requirements and other information that can affect your installation. The small amount of time that this review takes can save a much greater amount of time required to track down causes of installation errors later. Also, verify that you have the most current version of this document; Oracle documentation is updated after the initial release and posted to the following web site:

http://www.oracle.com/technetwork/indexes/documentation/index.html

Oracle recommends that you install a Web browser on your cluster nodes, both to enable Oracle Enterprise Manager and Oracle Application Express, and to access online documentation as needed. Online documentation is available in PDF and HTML formats.

> **See Also:**
>
> - *Oracle Database Concepts* for an overview of Oracle Database
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for additional information about Oracle Clusterware or Oracle RAC configuration and deployment
>
> - Procedure to clone an existing pluggable database (PDB) using Oracle ACFS snapshot at the following location:
>
>   https://www.youtube.com/watch?v=rhTTR9Jn43o

1.1.6 Select a Security Notification Contact

Oracle issues security alerts as needed for vulnerability fixes that are determined to be too critical to wait for distribution in the next Critical Patch Update.

During installation, you are asked in the Configure Security Updates screen to provide a security contact. Select one of the following options:

- Provide an e-mail address to receive security information for your installation.

- Provide a My Oracle Support e-mail address or account name to receive security information for your installation, and to enroll your system for Security Updates. You can receive information about alerts through My Oracle Support.

You can choose not to provide this information, but Oracle strongly recommends that you configure a security notification contact.

The information collected by Security Updates is limited to configuration information. The data collected does not include personally identifiable information (with the exception of a local contact name in case of transmission problems). You may still use all licensed Oracle functionality if you decline to enable Security Updates.

If you prefer not to receive security notifications, then leave all fields in the Configure Security Updates screen blank, and click **Next** to continue.

If you provide your My Oracle Support credentials, then Security Updates automatically gathers configuration information regarding your installed Oracle products and uploads it to Oracle's support systems. You can access the information it collects through your My Oracle Support account, and review health check recommendations, patch recommendations and other recommendations for your system in addition to security alerts.

> **See Also:** The Oracle Security Policies page, which is available from the following URL:
>
> http://www.oracle.com/us/support/assurance/fixing-policies/index.html

1.1.7 Review Root Task Delegation Options With Your System Administrator

During installation, Oracle Universal Installer (OUI) requires you to run scripts with superuser (or `root`) privileges to complete many system configuration tasks.

1.1.8 Review Installation Fixup Scripts With Your System Administrator

During an installation, OUI detects when the minimum requirements for an installation are not met, and creates shell scripts, called fixup scripts, to finish incomplete system configuration steps. If OUI detects an incomplete task, then it generates fixup scripts (`runfixup.sh`). You must run these scripts as `root`.

You can run the fixup script after you click **Fix and Check Again**. The fixup script modifies both persistent parameter settings and parameters in memory, so you do not need to restart the system, and can proceed with the installation.

1.1.9 Review How to Use Cluster Verification Utility (CVU)

You can use CVU before running OUI to ensure that your cluster is prepared for an Oracle RAC installation. CVU is incorporated into the installer, so CVU runs when you start an Oracle RAC installation. However, you can use CVU to ensure that any packages or configuration information that is required for Oracle RAC are in place before you begin your Oracle RAC installation.

Oracle provides CVU to perform system checks in preparation for an installation, patch updates, or other system changes. In addition, CVU can generate fixup scripts, which are scripts run by the `root` user that can change many kernel parameters to at least the minimum settings required for a successful installation.

Using CVU can help system administrators, storage administrators, and database administrators to ensure that everyone has completed the system configuration and preinstallation steps, so that installations, updates, or patches complete successfully. You can obtain the latest version of CVU at the following URL:

http://www.oracle.com/technetwork/database/options/clustering/downloads/cvu-download-homepage-099973.html

If you have a vendor who is performing hardware or operating system configuration steps, then ask the vendor to complete the relevant CVU check of the cluster after they complete their work to ensure that your system is configured correctly. Database administrators should refer to the section "Confirming Cluster Readiness for Oracle RAC Installation with CVU" on page 1-18 to confirm that their system is prepared for installation before they start an Oracle RAC installation.

1.1.10 Review Existing Oracle Installations and Upgrade Plans

If you have an existing Oracle installation, then document version numbers, patches, and other configuration information, and review upgrade procedures for your existing installation. Review the Oracle upgrade documentation before proceeding with installation, to decide how to proceed.

To install Oracle RAC 12*c* Release 1 (12.1), you must have Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) 12*c* Release 1 (12.1) installed on your cluster.

The Oracle Clusterware version must be equal to or greater than the Oracle RAC version that you plan to install.

> **See Also:** *Oracle Grid Infrastructure Installation Guide*

For the most current updates and best practices about pre-upgrade, post-upgrade, compatibility, and interoperability discussions, refer to "Oracle Upgrade Companion" for this release. "Oracle Upgrade Companion" is available through Note 1462240.1 on My Oracle Support:

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1462240.1

For upgrades, note the following:

- You can have only one version of Oracle Clusterware running on a cluster at a time. The Oracle Clusterware version must be the most recent release of any software (Oracle Clusterware, Oracle Database, Oracle RAC and Oracle Automatic Storage Management) running on the cluster. Before upgrading your database to Oracle Database 12*c* Release 1 (12.1), all nodes in the cluster must be upgraded to Oracle Clusterware and Oracle ASM 12*c* Release 1 (12.1).

- You can have multiple Oracle homes for Oracle Databases on your cluster. However, the Oracle RAC database software in these homes must be from a version that is equal to or before the version of Oracle Clusterware that is installed; you cannot have a version of Oracle Database running on Oracle Clusterware that was released after the version of Oracle Clusterware that you are running.

  For example:

  - If your servers are running Oracle Grid Infrastructure 12*c* Release 1 (12.1), then you can have an Oracle Database 12*c* Release 1 (12.1) single-instance database running on one node, and separate Oracle RAC 11g Release 1 or 2, or Oracle RAC 12*c* Release 1 (12.1) databases also running on the cluster.

  - You cannot have Oracle Grid Infrastructure 11*g* Release 2 installed on your cluster, and install Oracle RAC 12*c* Release 1 (12.1).

- You can use Database Upgrade Assistant (DBUA) for patch set upgrades with Oracle RAC. You can also use DBUA to upgrade between major point releases of Oracle RAC. For example, you can upgrade from Oracle Database 11*g* Release 2 (11.2) to Oracle Database 12*c* Release 1 (12.1).

- You cannot change the owner of the Oracle Database home during an upgrade. You must use the same Oracle software owner that owns the existing Oracle Database home.

- If you are upgrading an existing Oracle RAC installation, then you must use the same type of Oracle home that you have in your existing installation. For example, if you have a shared Oracle home in your existing installation, then you must upgrade to a shared Oracle home with Oracle RAC 12*c* Release 1 (12.1).

  Similarly, if you have local Oracle homes on cluster nodes, then you must upgrade to local Oracle homes on cluster nodes.

- For the Oracle RAC database being upgraded, each cluster member node that hosts an instance of the database must be a member of the upgraded installation. For example, if you have an existing Oracle RAC database running on a three-node cluster, then you must upgrade Oracle RAC on all three nodes. You cannot upgrade only two nodes of the cluster, and then remove the third node and instance in the upgrade.

- If you have any database data stored on RAW devices, then before you start Oracle Grid Infrastructure and Oracle RAC installation, you must use RMAN to copy that data to Oracle ASM or to another supported file system.

- As with any system change, follow common industry standards for data recovery planning, and back up your existing database before upgrading your software.

> **See Also:** *Oracle Database Upgrade Guide*

1.1.11 Consider Memory Allocation and Automatic Memory Management

During a Typical installation, you create your database with Database Configuration Assistant (DBCA), and automatic memory management is enabled. If you choose advanced installation, then you can either specify memory allocation manually, or enable automatic memory management.

With automatic memory management, the Oracle RAC instances automatically manage and tune memory for you. With automatic memory management, you choose a memory target, and the instance automatically distributes memory between the system global area (SGA) and the instance program global area (instance PGA). As memory requirements change, the instance dynamically redistributes memory between the SGA and instance PGA.

If you choose not to enable automatic memory management during installation, then you can enable it after installation. Enabling automatic memory management after installation involves a shutdown and restart of the database.

Starting with Oracle Database 12*c*, Oracle Database uses the Optimized Shared Memory (OSM) feature of Oracle Solaris, instead of Dynamic Intimate Shared Memory (DISM), to implement Automatic Memory Management. Do not manually enable or use DISM to implement shared memory on Oracle Solaris.

> **Note:** Oracle Database uses shared memory extensively. See *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems* for information about tuning your Oracle Solaris shared memory environment.

> **See Also:** *Oracle Database Administrator's Guide* for more information about automatic memory management

1.1.12 Review Globalization Requirements

Before you start an installation on which you want to support languages other than English, review *Oracle Database Globalization Support Guide*.

Note the following:

- Oracle recommends that you use Unicode AL32UTF8 as the database character set.

  Unicode is the universal character set that supports most of the currently spoken languages. Unicode also supports many historical scripts (alphabets). Unicode is the native encoding of many technologies, including Java, XML, XHTML, ECMAScript, and LDAP. Unicode is ideally suited for databases that support the Internet and the global economy.

- The locale setting of your operating system session determines the language of the user interface and the globalization behavior for components such as Oracle Universal Installer, Oracle Net Configuration Assistant, and Database

Configuration Assistant. The locale setting also determines the globalization behavior of Oracle Database sessions created by a user application through Oracle JDBC driver, unless overridden by the application.

■ The character set specified by the `NLS_LANG` environment variable determines the language of the user interface and the globalization behavior for components such as SQL*Plus, `exp`, and `imp`. This variable also sets the language and territory used by the client application and the database. The variable also declares the character set for entering and displaying data by the client application.

The `NLS_LANG` environment variable should reflect the setting of the operating system character set of the client. For example, if the database character set is AL32UTF8 and the client is running on a Windows operating system, then you should not set the `NLS_LANG` environment variable to AL32UTF8 because there are no UTF-8 WIN32 clients. Instead, the `NLS_LANG` setting should reflect the code page of the client. For example, on an English Windows client, the code page is 1252, so an appropriate setting for `NLS_LANG` is `AMERICAN_AMERICA.WE8MSWIN1252`.

> **Note:** *Oracle Database Installation Guide* for your platform contains a more complete discussion of database character sets used with different languages, and provides further information about installing and configuring Oracle Database globalization support.

1.2 Overview of Server Hardware, Network, and Operating System

Oracle Clusterware must be installed successfully as part of an Oracle Grid Infrastructure for a cluster installation before attempting to install Oracle RAC. If you plan to install Oracle RAC for an Oracle Flex Cluster, then you must have installed Oracle Clusterware using the Oracle Flex Cluster option.

To complete installations successfully, ensure that required hardware, network, and operating system preinstallation steps for Oracle software are performed as required. Failure to complete the required preinstallation steps is the most common reason for unsuccessful installations.

Before Oracle Clusterware is installed as part of an Oracle Grid Infrastructure for a cluster installation, you must have completed installing and configuring CPUs, memory, shared storage, local disks, network cards, host bus adapters, interconnects, and any other networking or server hardware. You should also have installed the operating system, and any required packages or third-party software. Review your vendor documentation to complete these tasks, and if relevant, work with your vendor to complete the Oracle preinstallation steps that are listed in this document to confirm that the vendor hardware and software is correctly configured.

Server and network preparation for installation includes the following:

■ Review Server Hardware and Software Requirements

■ Overview of Server Environment Configuration

■ Overview of Oracle Base and Oracle Home Directories

■ Overview of Network and Internet Configuration

■ Overview of Platform-Specific Server Configuration

1.2.1 Review Server Hardware and Software Requirements

The following summary of server hardware and software configuration requirements and recommendations will enable you to prepare for a successful installation of Oracle RAC.

1.2.1.1 Cluster Node Member Requirements for Oracle RAC

Each node in a cluster requires the following:

- Supported server hardware, including processors and system configuration.

  Review My Oracle Support before starting an installation on your current hardware and before purchasing new hardware to ensure that the hardware is supported by Oracle Clusterware with Oracle RAC 12*c* Release 1 (12.1).

  Also review the preinstallation chapters in *Oracle Grid Infrastructure Installation Guide* for more details about the supported configurations.

  > **Note:** You must use the same operating system on each node in the cluster. Oracle strongly recommends that you use the same software configurations on each node of the cluster. Oracle Clusterware and Oracle RAC do not support heterogeneous platforms (each server must run the same Oracle software binaries) in the same cluster.

- Operating system updates listed in the system requirements.

  For some operating systems, Oracle may require updates, such as service packs and individual patches. If such requirements exist, then they are stated in the Release Notes for a particular release. Refer to your operating system vendor for required operating system updates.

- Network connections required for the node type you are configuring (Hub Node or Leaf Node).

1.2.1.2 Recommended for Oracle RAC: System Administrator Tasks

Oracle recommends the following tasks to simplify server installation and maintenance, and to prevent service issues:

- Using a time protocol, such as Network Time Protocol (NTP), to ensure that all nodes in the cluster use the same reference time. With Oracle Clusterware 12*c* Release 1 (12.1), if NTP servers are not provided during Oracle Grid Infrastructure installation, then the Oracle Clusterware installation enables the Cluster Time Synchronization Service (CTSS).

- Configuring redundant switches, for all cluster sizes.

- Using identical server hardware on each node, to simplify server maintenance.

- Avoiding resource contention issues by not installing Oracle RAC on a primary domain controller or backup domain controller.

1.2.1.3 Additional Options: System Administrators and Vendors

After you have set up server hardware, review "Checking the Hardware Requirements" in *Oracle Grid Infrastructure Installation Guide* for your platform to ensure that your system has enough RAM, that the TEMP environment variable points to a location that has enough available space for the installation, and that your system meets other hardware requirements.

1.2.2 Overview of Server Environment Configuration

Configure the users and user environments as described in the preinstallation chapters of *Oracle Grid Infrastructure Installation Guide*. These include the following tasks:

- Creating operating system users to install Oracle software
- Configuring the Oracle software owner user environments

1.2.3 Overview of Oracle Base and Oracle Home Directories

The Oracle base directory is the location where Oracle configuration files installed by a particular Oracle installation owner are stored. An Oracle base directory can be used for multiple installations of software by a given installation owner.

The Oracle base and Oracle home directory structures are different for Oracle Database and Oracle Grid Infrastructure for a cluster. By default, the Oracle Database software binary files are installed by version and Oracle home name in a subdirectory of the Oracle base directory. Every Oracle home can have its own Oracle base. The Oracle Grid Infrastructure for a cluster software binaries are stored in an Oracle home (the Grid home) that is outside of the Grid Infrastructure installation owner Oracle base. There can be only one active Grid home on each cluster member, or only one active shared home, and it must not be under the Oracle base for the Grid Infrastructure installation owner.

Typically, the Oracle base path is based on the installation owner name. If you have separate installation owners for the Oracle Grid Infrastructure installation and the Oracle RAC installation, then you have two Oracle base paths that are in accordance with Optimal Flexible Architecture (OFA) guidelines. For example, if the user `grid` is the installation owner of the Oracle Grid infrastructure installation and the user `oracle` is the installation owner of the Oracle Database installation, then you have two Oracle base directories:

- `u01/app/grid`: This is the Oracle base for the Grid installation owner (`grid`), which owns the Oracle Grid Infrastructure binaries
- `u01/app/oracle`: This is the Oracle base for the Oracle Database installation owner (`oracle`), which owns the Oracle Database binaries

In addition, you have a Grid home that is in a different path than the Oracle base path for the Grid installation owner, and the Oracle base path for the Oracle Database owner:

`/u01/app/12.1.0/grid`

The Grid home path is outside of the Oracle base directory for both the Oracle Database installation owner (`oracle`) and the Oracle Grid Infrastructure owner (grid). The path to the grid home (`/u01` and `/u01/app`) is owned by `root` after installation, so the Oracle base for installation owners should be created by system administrators, and given read, write and execute permissions to the Oracle installation owner that is permitted to write to that Oracle base.

Oracle recommends that you do not set an Oracle home environment variable, and instead enable OUI to create it. If the Oracle base path is `/u01/app/oracle`, then by default, OUI creates the following Oracle home path:

`/u01/app/oracle/product/12.1.0/dbhome_n`

The variable *n* is the Oracle home number. The first time you create an Oracle home as the `oracle` user, the default Oracle home location is the following:

`u01/app/oracle/product/12.1.0/dbhome_1`

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

1.2.4 Overview of Network and Internet Configuration

You must configure each server so it can communicate with the other nodes in the cluster. Also, you should configure one or more nodes to access the Internet and support a web browser.

1.2.4.1 Overview of Network Interface Configuration

During the installation of Oracle Grid Infrastructure for a cluster, identify the planned use for global interface for each node, and provide the interface information required for the type of node (Hub Node or Leaf Node) that you are configuring.

Identify each interface as one of the following:

- A *Public* interface type (used with public IP addresses and virtual IP addresses)

- A *Private* interface type (used with the networks for the interconnect between cluster member nodes)

- An *ASM* interface type (used for communication with Oracle ASM instances on other nodes)

- An *ASM and Private* interface type (used for both interconnects between cluster member nodes and communication with Oracle Flex ASM instances on other nodes)

- A *Do not use* interface type, which Oracle Clusterware and Oracle RAC ignores

For example, an interface used as a dedicated interface for a network file system such as NFS should be marked as a *Do not use* interface type.

You configure the network during the Oracle Grid Infrastructure installation; you do not perform additional network configuration during an Oracle RAC installation.

> **See Also:** *Oracle Grid Infrastructure Installation Guide* for detailed information about network requirements

1.2.4.2 Overview of Browser Requirements

Web browsers are required to use Oracle Enterprise Manager Database Express and Oracle Enterprise Manager Cloud Control. Web browsers must support JavaScript, and the HTML 4.0 and CSS 1.0 standards. For a list of browsers that meet these requirements, see the Oracle Enterprise Manager certification matrix on My Oracle Support:

https://support.oracle.com

> **See Also:** *Oracle Enterprise Manager Cloud Control Basic Installation Guide* for information on accessing the Oracle Enterprise Manager certification matrix

1.2.5 Overview of Platform-Specific Server Configuration

Oracle Clusterware and Oracle RAC are tested with specific operating system versions, and specific operating system components. Oracle requires that you use the operating system versions and components that are certified for this release.

Oracle recommends that you or your system administrator review the system requirements carefully in *Oracle Grid Infrastructure Installation Guide* before beginning installation, to ensure that your system meets these requirements. If your system does not meet minimum hardware, operating system version, and component requirements, then your installation may fail to complete, or other errors may develop during Oracle Clusterware or Oracle Database run time.

In addition to the standard system requirements configuration, deployment on specific server hardware can include additional operating system configuration steps. Review the preinstallation chapter in *Oracle Grid Infrastructure Installation Guide*, and check the My Oracle Support Certify page to ensure that you are aware of any additional requirements or recommendations for your specific hardware and platform configuration.

1.3 Overview of Oracle RAC Storage

The hard disk requirements for Oracle Database components include 32 MB required to install Java Runtime Environment (JRE) and OUI on the disk partition where the operating system is installed. If sufficient space is not detected, then the installation fails and an error message appears.

To install Oracle RAC, you must configure shared storage for the database files and optionally the recovery files. The database fast recovery area must be located on shared storage.

See *Oracle Grid Infrastructure Installation Guide* to review storage options for installation planning. Storage and system administrators can see this chapter to configure storage for database files for an Oracle RAC database.

> **See Also:** *Oracle Database Administrator's Guide* for information about recovery areas

1.3.1 About Oracle RAC Storage and Oracle ACFS

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a multiplatform, scalable file system, and storage management technology that extends Oracle ASM functionality to support customer files maintained outside of the Oracle Database. Files supported by Oracle ACFS include database and application executable files, trace files, alert logs, application reports, PFILEs, and configuration files.

1.3.1.1 About the Oracle ASM Dynamic Volume Manager

The Oracle ASM Dynamic Volume Manager (Oracle ADVM) provides volume management services for disks and a standard disk device driver interface to clients (users and applications). File systems and other disk-based applications issue I/O requests to Oracle ADVM volume devices as they would to other storage devices on a vendor operating system. Oracle ADVM extends Oracle ASM by providing a disk driver interface to Oracle ASM storage allocated as Oracle ASM volume files. You can use Oracle ADVM to create virtual disks that contain file systems. These file systems contained on Oracle ASM volumes can support files beyond Oracle database files,

such as executable files, report files, trace files, alert logs, and other application data files.

1.3.2 General Storage Considerations for Oracle RAC

For all installations, you must choose the storage option to use for Oracle RAC database files. You do not have to use the same storage option for each file type.

Use the following guidelines when choosing the storage options to use for each file type. You can choose any combination of the supported shared storage options for each file type if you satisfy all requirements listed for the chosen storage options

- Oracle recommends that you choose Oracle ASM as the shared storage option for database and recovery files. You do not have to have an Oracle ASM instance running on every hub node in a Big Cluster configuration, but every hub node must have direct access to the shared storage.

- For Standard Edition and Standard Edition 2 (SE2) Oracle RAC installations, Oracle ASM is the only supported shared storage option for database or recovery files.

- If you do not have a storage option that provides external file redundancy, then you must configure at least three voting disk areas to provide voting disk redundancy.

- If you intend to use Oracle ASM with Oracle RAC and you did not use Oracle ASM for the Oracle Clusterware files (Oracle Cluster Registry (OCR) and voting disk), then you must configure one or more Oracle ASM instances with Oracle ASM Configuration Assistant (ASMCA) before starting OUI. Oracle ASM runs on one ore more hub nodes in the cluster.

- If you intend to upgrade an existing Oracle RAC database, then you must ensure that your system meets the following conditions:

    - OUI and DBCA are run on the node where the Oracle RAC database instance is located.

    - The Oracle RAC database instance is running on the *same* nodes that you intend to make members of the new cluster installation. For example, if you have an existing Oracle RAC database running on a three-node cluster, then you must install the upgrade on all three nodes. You cannot upgrade only two nodes of the cluster, removing the third instance in the upgrade.

        See Also: *Oracle Database Upgrade Guide* for information about how to prepare for upgrading an existing database

1.3.3 Overview of Oracle Database and Recovery File Options

There are two ways of storing Oracle Database and recovery files:

> **Note:** When creating a new Oracle RAC database using OUI, storing data files and recovery files on raw devices (unformatted partitions) is not supported. Place data files and recovery files on a shared file system or on Oracle ASM. If you are upgrading a database that uses raw devices, then they are still supported with Oracle Database 11*g* Release 2.

■ **Oracle Automatic Storage Management**: Oracle ASM is an integrated, high-performance database file system and disk manager for Oracle Database files. It performs striping and mirroring of database files automatically.

> **Note:** For Standard Edition and Standard Edition 2 (SE2) Oracle Database installations using Oracle RAC, Oracle ASM is the only supported storage option.

■ **A supported shared file system**: Supported file systems include the following:

– **Direct Network File Systems Client (Direct NFS Client):** You can configure Oracle Database to access NFS servers directly using Direct NFS Client. Direct NFS Client supports using NFSv3, NFSv4 and NFSv4.1 (excluding Parallel NFS) to access the NFS server. Direct NFS Client is an optimized NFS client that provides faster and more scalable access to NFS storage located on NAS storage devices (accessible over TCP/IP). Direct NFS Client is built directly into the database kernel and provides faster performance than what can be provided by the operating system's NFS driver. When accessing NFS servers, Oracle Database bypasses the operating system and generates exactly the requests it needs (no user configuration or tuning required).

Table 1–1 shows the storage options supported for storing Oracle Database files and Oracle Database recovery files. Oracle Database files include data files, control files, redo log files, the server parameter file (SPFILE), and the password file.

Table 1–1   Supported Storage Options for Oracle Database and Recovery Files

Storage Option	File Types Supported	
	Database	**Recovery**
Oracle ASM	Yes	Yes
Oracle ASM Cluster File System (Oracle ACFS)	Yes	Yes
Direct NFS Client	Yes	Yes
Local storage	No	No
Shared unformatted partitions	No	No

1.4 Additional Information to Prepare for Installation

This section contains additional information about Oracle Clusterware, Oracle Automatic Storage Management (Oracle ASM), and Oracle RAC, that may be helpful for your installation plan team to read to decide how to configure your installation. It contains the following topics:

■ Oracle ASM is Installed with Oracle Grid Infrastructure

■ Hub Nodes and Shared Components of Oracle RAC Installations

■ Simplified Upgrade of TIMESTAMP WITH TIME ZONE Data

■ Ensure an External Jobs User Is Created for HP-UX

■ Oracle Database 12c Real Application Clusters Release Compatibility

1.4.1 Oracle ASM is Installed with Oracle Grid Infrastructure

In past releases, Oracle ASM was installed as part of the Oracle Database installation. With Oracle Database 11*g* Release 2 (11.2) and later releases, Oracle ASM is part of an Oracle Grid Infrastructure installation. To upgrade an existing Oracle ASM installation, you must upgrade Oracle ASM by running an Oracle Grid Infrastructure upgrade.

If you did not select Oracle ASM as the storage option for the Oracle Clusterware files, then you must first use Oracle ASM Configuration Assistant (ASMCA) on one of the Hub Nodes to configure Oracle ASM, create an Oracle ASM instance, and create a disk group to use for your Oracle Database storage.

1.4.2 Hub Nodes and Shared Components of Oracle RAC Installations

For Oracle RAC, you and your system administrator should note that all Hub Node instances in Oracle RAC environments share the control file, server parameter file, redo log files, and all data files. These files must be placed on Oracle ASM, and all the cluster database instances on the Hub Node must have access to these files. 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.

> **See Also:** *Oracle Automatic Storage Management Administrator's Guide* for more information about Hub Nodes

1.4.3 Simplified Upgrade of TIMESTAMP WITH TIME ZONE Data

As part of an installation of Oracle Database 12*c* Release 1 (12.1), time zone version files from 1 to 12 are installed in the path `$ORACLE_HOME/oracore/zoneinfo/`. You can continue to use the current time zone version or upgrade to the latest version. Oracle recommends that you upgrade the server to the latest time zone version. Upgrading to a new time zone version may cause existing TIMESTAMP WITH TIME ZONE (TSTZ) data to become stale. Using the newly provided DBMS_DST PL/SQL packages, you can update the TSTZ data transparently with minimal manual procedures and system downtime.

All instances of an Oracle RAC database must use the same time zone. The Oracle RAC database time zone defaults to the time zone setting of the Oracle Grid Infrastructure Grid user, except when an instance is started with SQL*Plus. When you use SQL*Plus, you must be sure to use the same time zone setting for the database instance that is used for Oracle Clusterware. You can change the time zone Oracle Clusterware uses for a database by using the following command, where `time_zone` is the time zone to which you want to change:

```
srvctl setenv database -env 'TZ=time_zone'
```

Time zone version files are also installed on the clients. Starting with Oracle Database 11*g* Release 2, you no longer need to upgrade Client time zone files immediately. Upgrades can be done at a time when it is most convenient to the system administrator. However, there could be a small performance penalty when client and server use different time zone versions.

See Also:

- *Oracle Database Upgrade Guide* for information about preparing to upgrade TSTZ data

- *Oracle Database Globalization Support Guide* for information about how to upgrade the time zone file and TSTZ data

- *Oracle Call Interface Programmer's Guide* for information about performance effects of clients and servers operating with different versions of time zone files

1.4.4 Ensure an External Jobs User Is Created for HP-UX

If you are installing Oracle RAC on HP-UX, then ensure that there is a low-privileged user created to own external jobs. Be aware that you must set this user as the external jobs user by logging in as the `root` user and modifying `$ORACLE_HOME/rdbms/admin/externaljob.ora` after installation. Only the `root` user can modify this file.

1.4.5 Oracle Database 12*c* Real Application Clusters Release Compatibility

You can install and operate different releases of Oracle Database software on the same computer:

- If you have Oracle Clusterware installed and different releases of other Oracle software installed, then the Oracle Clusterware release must be later than or equal to the Oracle Database software release. Oracle Clusterware and Oracle ASM are both upgraded to 12*c* Release 1 (12.1) when you perform an Oracle Grid Infrastructure 12*c* Release 1 (12.1) installation

- If you have an existing Oracle home, then you can create a new Oracle home and install Oracle Database 12*c* Release 1 (12.1) into the new Oracle home. You should ensure that Oracle Clusterware is in a separate Oracle Grid Infrastructure home. Oracle Grid Infrastructure for a Cluster installations cannot be installed in the Oracle base directory for Oracle Database.

- If you are running Oracle Database 9*i* Release 2 Oracle RAC, and you want to continue to use that release, then you must run cluster software that is compatible with that release, such as Oracle Cluster Manager or a third party cluster software.

  Note: To remove third party cluster software after upgrading your database, you must first remove the third party cluster software, and then re-install Oracle Clusterware.

- If OUI detects a previous database release, then OUI asks you about your upgrade preferences. You have the option to upgrade one of the previous release databases with DBUA or to create a database using DBCA. The information collected during this dialog is passed to DBUA or DBCA after the software is installed.

  If OUI detects a previous Oracle Clusterware release, then you are asked to upgrade the existing Oracle Clusterware installation. Only one Oracle Clusterware version can be active on a server, and a server must be a member of only one cluster.

> **Note:** Do not move Oracle executable files from the directory in which they were installed to another location. Doing so can cause dynamic link failures. To move the executable files to a new location, you must reinstall the software.

- You cannot install Oracle Grid Infrastructure for a standalone server (Oracle Restart) and then install Oracle Grid Infrastructure for a cluster. If you have Oracle Restart installed, then you must remove that installation before you can install Oracle Grid Infrastructure for a cluster.

- You can run different releases of Oracle Database and Oracle ASM. For example, you can use Oracle ASM 12*c* Release 1 (12.1) with an Oracle Database 11*g* Release 2 (11.2) database. However, you can only run an Oracle Database version of a release before Oracle Grid Infrastructure 12*c* on a Hub Node. You cannot run earlier releases on a Leaf Node.

> **Note:** When using different Oracle ASM and Oracle Database releases, the functionality of each depends on the functionality of the earlier software release. For example, an Oracle Database 11*g* Release 11.2 database using an Oracle ASM 12*c* Release 1 (12.1) instance is not able to use new features available for Oracle ASM 12*c* Release 1 (12.1), but instead it can use only Oracle ASM 11.2 features.

1.5 Confirming Cluster Readiness for Oracle RAC Installation with CVU

Before you start your installation, use Cluster Verification Utility (CVU) to ensure that your system is prepared for Oracle RAC installation. If any checks fail, then fix the errors reported, either manually or by using a generated fixup script, or contact your system or storage administrator to have the cause of the errors addressed.

CVU is available in the Grid home, in the `bin` directory. For example, if the Oracle Grid Infrastructure for a cluster home is `/u01/app/12.1.0/grid`, then the path is `/u01/app/12.1.0/grid/bin`. To start CVU, navigate to the Grid home `bin` directory, and use a command similar to the following:

```
cluvfy stage -pre dbinst -fixup -n nodelist -r release -osdba OSDBA -verbose
```

In the preceding command syntax, *nodelist* is a comma-delimited list of node names and *release* is the version of the Oracle Database software being installed. The `-fixup`, `-osdba`, and `-verbose` flags are optional.

For example, for a two-node cluster with nodeA and nodeB, where you are testing the cluster to prepare to install Oracle Database 12*c* with Oracle RAC, and your OSDBA group is `dba`, the following command checks for system readiness:

```
$ ./cluvfy stage -pre dbinst -fixup -n nodea,nodeb -osdba dba -verbose
```

For more information about CVU commands, enter `./cluvfy -help`.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for detailed information about CVU

1.6 Installing Oracle Database 12*c* on Systems with Existing Databases

If you are planning an installation on a system where you have an existing Oracle RAC or Oracle Database installation, then you must perform additional tasks to prepare your system for installation.

Table 1–2 provides an overview of what you must do if you have an existing Oracle Database installation. Review the table, and perform tasks as required.

> **See Also:** *Oracle Database Upgrade Guide* for additional information about preparing for and performing upgrades

Table 1–2   Overview of System Preparation for Upgrades or Coexisting Databases

Installation Scenario	What you must do
Upgrading from Oracle Database 10*g* Release 1 (10.1) to 12*c* Release 1 (12.1)	No additional tasks. Refer to About Oracle Database 10g Listener Migration on page 1-19
Installing Oracle Database 12*c* Release 1 (12.1) on a system to coexist with Oracle Database 10*g* Release 1 (10.1)	No additional tasks. Refer to About Oracle Database 10g Listener Migration on page 1-19

1.7 About Oracle Database 10*g* Listener Migration

If your system has an Oracle Database 10*g* installation, and you install Oracle Database 12*c* Release 1 (12.1) either to coexist with or to upgrade the Oracle Database 10*g* installation, then most installation types configure and start a default Oracle Net listener using TCP/IP port 1521 and the IPC key value EXTPROC. One of the following occurs:

- During a coexisting installation, Database Configuration Assistant (DBCA) automatically migrates the listener and related files from the Oracle Database 10*g* Oracle home to the Oracle Database 11*g* Oracle home.

- During an upgrade, Oracle Database Upgrade Assistant (DBUA) automatically locates the existing Oracle Database 10*g* listener, and migrates the listener to Oracle Database 12*c* Release 1 (12.1).

> **Note:** For Oracle Database 12*c* Release 1 (12.1) to coexist with Oracle Database 10*g* Release 1 (10.1) databases using Oracle ASM, the database must be Oracle Database 10*g* Release 10.1.0.3 or later.

2

Installing Oracle RAC and Oracle RAC One Node

This chapter describes phase two of the installation procedures for installing Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node databases. Phase one is the completion of installing Oracle Clusterware, as described in *Oracle Grid Infrastructure Installation Guide* for your platform.

> **See Also:** *Oracle Database Installation Guide* for your platform for additional information, such as how to open compressed files, and other information about setting up files for installation
>
> Also, during installation, click **Help** to find explanations of options and prompts for each installation screen, and click **Details** to see the log file

This chapter contains the following topics:

- Selecting a Security Notification Contact
- Deciding Between Multitenant Container Databases and Non-CDBs in Oracle RAC
- Selecting an Installation Option
- Selecting the Database Type for Oracle Grid Infrastructure Deployments
- Choosing the Cluster Database Management Type
- Selecting an Installation Type
- Selecting an Oracle Database Configuration Type
- Selecting a Database Name
- Creating Database Passwords
- Understanding the Actions of OUI and DBCA During Installation
- Using Oracle Universal Installer to Install Oracle RAC
- Using Oracle Universal Installer to Install Oracle RAC One Node
- Installing the Oracle Database Vault Option

2.1 Selecting a Security Notification Contact

Oracle issues security alerts as needed for vulnerability fixes that are determined to be too critical to wait for distribution in the next Critical Patch Update.

During installation, you are asked in the Configure Security Updates screen to provide a security contact. Select one of the following options:

- Provide an e-mail address to receive security information for your installation.

- Provide a My Oracle Support e-mail address or account name to receive security information for your installation, and to enroll your system for Security Updates. You can receive information about alerts through My Oracle Support.

You can choose not to provide this information, but Oracle strongly recommends that you configure a security notification contact.

The information collected by Security Updates is limited to configuration information. The data collected does not include personally identifiable information (with the exception of a local contact name in case of transmission problems). You may still use all licensed Oracle functionality if you decline to enable Security Updates.

If you prefer not to receive security notifications, then leave all fields in the Configure Security Updates screen blank, and click **Next** to continue.

If you provide your My Oracle Support credentials, then Security Updates automatically gathers configuration information regarding your installed Oracle products and uploads it to Oracle's support systems. You can access the information it collects through your My Oracle Support account, and review health check recommendations, patch recommendations and other recommendations for your system in addition to security alerts.

> **See Also:** The Oracle Security Policies page, which is available from the following URL:
>
> http://www.oracle.com/us/support/assurance/fixing-policies/index.html

2.2 Deciding Between Multitenant Container Databases and Non-CDBs in Oracle RAC

Starting in Oracle Database 12*c* Release 1 (12.1), you must create a database as either a multitenant container database (CDB) or as an Oracle database that is non-CDB. This also applies to Oracle RAC databases. The only difference to the installation process is to choose whether to create the Oracle RAC database as a CDB or non-CDB.

A pluggable database (PDB) is a portable collection of schemas, schema objects, and nonschema objects that appears to an Oracle Net client as a non-CDB. PDBs can be plugged into CDBs. A CDB can contain multiple PDBs. Each PDB appears on the network as a separate database.

If you create an Oracle RAC database as a CDB and plug one or more PDBs into the CDB, then, by default, a PDB is not started automatically on any instance of the Oracle RAC CDB. With the first dynamic database service assigned to the PDB (other than the default database service which has the same name as the database name), the PDB is made available on those instances on which the service runs.

Whether or not a PDB is available on more than one instance of an Oracle RAC, CDB is typically managed by the services running on the PDB. You can manually enable PDB access on each instance of an Oracle RAC CDB by starting the PDB manually on that instance.

> **See Also:**
>
> - *Oracle Database Concepts* for more information about PDB concepts
>
> - *Oracle Database Administrator's Guide* for more information about managing PDBs
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for information specific to the administration of Oracle RAC CDBs

2.3 Selecting an Installation Option

On the Select Installation Option page, you are given the following options:

- **Create and Configure a Database**: Provides you with the option to create a database using a preconfigured database template designed for particular system load demands, such as an online transaction processing (OLTP) database, or a decision support or data warehouse database.

- **Install database software only**: Installs Oracle Database software; you must complete the database configuration after the installation completes using the installed utilities.

- **Upgrade an existing database**: Upgrades an existing database.

If you are installing Oracle Database software, then Oracle recommends that you use a preconfigured database option, or select the **Advanced** option on the Select Configuration page, and configure a custom starter database. See "Selecting an Installation Type" for more information about the different preconfigured database options.

If you have an existing Oracle installation, then write down the version numbers, patches, and other configuration information, and review upgrade procedures for your existing installation. Review *Oracle Database Upgrade Guide* before proceeding with the installation.

> **See Also:** For the most recent updates and best practices about pre-upgrade, post-upgrade steps, compatibility, and interoperability discussions, see "Oracle Upgrade Companion," which is available through Note 1462240.1 on My Oracle Support:
>
> https://support.oracle.com/CSP/main/article?cmd=show&type=NO T&id=1462240.1

2.4 Selecting the Database Type for Oracle Grid Infrastructure Deployments

During installation, OUI detects if you have Oracle Grid Infrastructure for a cluster installed. If you do, then you must specify the type of database you plan to create:

- A single-instance database

- An Oracle RAC database

- An Oracle RAC One Node database

If you plan to create databases of different types on this cluster, then choose the most advanced option. For example, if you plan to create only single-instance and Oracle RAC One Node databases, then choose the Oracle RAC One Node database option. If

you plan to create single-instance databases and Oracle RAC databases, then choose the Oracle RAC database option.

You can install Oracle RAC One Node on two or more nodes in the cluster. An Oracle RAC One Node installation starts up an instance on one of the nodes you select as an Oracle RAC One Node pool member. If that instance goes down, then the Oracle RAC One Node instance fails over to another pool member. This feature relocates database instances and connections to other cluster nodes for high availability.

> **See Also:**
>
> - *Oracle Database High Availability Overview* for information about Oracle RAC One Node
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for information about how to convert single-instance databases to Oracle RAC

2.5 Choosing the Cluster Database Management Type

When creating an Oracle RAC database, you can choose to create on of the following types of databases:

- A **policy-managed databas**e: The database instances are automatically managed based on server pools for effective resource utilization.

- An **administrator-managed database**: The database instances are tied to specific servers in the cluster.

  > **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* for more information about server pools and the different cluster database management types

2.6 Selecting an Installation Type

When you run Oracle Universal Installer (OUI) to install Oracle RAC, you can select the **Typical** or the **Advanced** installation type.

The Typical installation type installs a default configuration of Oracle Database, with basic configuration choices. Oracle recommends that most users select Typical as their installation type.

The Advanced installation type is for customized installations. You should use Advanced installation only when you have a specific requirement for it, such as adding specific components to your installation, requiring different passwords for the SYS, SYSTEM and DBSNMP accounts, using a different database character set than is in use on your servers, changing product languages, or other nonstandard configurations.

If you plan to use any of the following options during installation, then you should choose the Advanced Installation Type:

- Preconfigured Database Types Supplied with Oracle Database

- Using Advanced Database Configuration

- Installing Oracle Database with Other Languages

2.6.1 Preconfigured Database Types Supplied with Oracle Database

The General Purpose and Transaction Processing type and the Data Warehouse type use preconfigured database templates optimized for each type of database.

During installation, OUI starts Oracle Net Configuration Assistant (NETCA) and DBCA, and installs the preconfigured database without further input. During database installation, OUI displays a progress indicator.

DBCA processing for these two configuration types creates a starter database, and configures the Oracle network services.

2.6.2 Using Advanced Database Configuration

Advanced configuration options available using this installation type include Oracle RAC, Automatic Storage Management, backup and recovery configuration, integration with Oracle Enterprise Manager Cloud Control, more fine-grained memory tuning, and other options.

2.6.3 Installing Oracle Database with Other Languages

To use languages other than the default (English), either for the database or for applications running on the database, you must use the Advanced Installation method.

> **See Also:**
>
> *Oracle Database Globalization Support Guide* for detailed information about character sets and language configuration

2.7 Selecting an Oracle Database Configuration Type

When you choose the Advanced Installation option in OUI, then you can select the **General Purpose/Transaction Processing**, **Data Warehouse**, or **Advanced** database configuration type. If you choose the Typical Installation Option, then the database configuration type defaults to General Purpose/Transaction Processing.

If you select Advanced configuration, then you can use Oracle Database Configuration Assistant (DBCA) to create the database as described in Chapter 2, "Installing Oracle RAC and Oracle RAC One Node". Oracle recommends that you use DBCA to create your database.

2.8 Selecting a Database Name

The database name input field sets the following Oracle initialization parameter values:

- `DB_NAME`

- `DB_UNIQUE_NAME`

- `DB_DOMAIN`

In Oracle RAC environments, the database name (`DB_UNIQUE_NAME`) portion is a string of no more than 30 characters that can contain alphanumeric, underscore (_), dollar ($), and pound (#) characters, but must begin with an alphabetic character. No other special characters are permitted in a database name. The `DB_NAME` parameter for a database is set to the first 8 characters of the database name.

The domain portion of the global database name (DB_DOMAIN) can be no more than 128 characters. Domain names using underscores (_) are not allowed. The values for DB_UNIQUE_NAME.DB_DOMAIN in its entirety must be unique within the enterprise.

For example, if your database has a global database name of orl$racprod2551.example.com which you supplied during installation, then the following values are used for initialization parameters:

```
DB_UNIQUE_NAME        orl$racprod2551

DB_DOMAIN             example.com

DB_NAME               orl$racp
```

The SID prefix is the first 8 characters of the database name. The SID prefix can contain only the characters a-z, A-Z, and 0-9. The SID prefix cannot contain operating system special characters, so if you use special characters in the first 8 characters of the database name, then these special characters are omitted in the SID prefix. There is a single SID prefix for every database. The SID prefix for a database must be unique within the cluster.

For an Oracle RAC database, each instance has a unique identifier, ORACLE_SID, which consists of the SID prefix and an instance number. The ORACLE_SID for Oracle RAC database instances is generated differently, depending on how you choose to manage the database. If you select a policy-managed database, then Oracle generates the SID in the format *name_#*, where name is the first eight alphanumeric characters of DB_UNIQUE_NAME, and # is the instance number. If you select an administrator-managed database, then DBCA generates the SID for the instance names in advance, and the SID is in the format *name#*.

For an Oracle RAC One Node database, the instance is ORACLE_SID_1, which consists of _1 appended to the SID prefix. During online relocation, a second instance ORACLE_SID_2 is started, which becomes the only instance after the relocation completes. The next online relocation uses ORACLE_SID_1 for the new instance.

For example, if the DB_UNIQUE_NAME for a database is orl$racprod2551, then the following SID values are used:

Database or Instance Type	Value Used for ORACLE_SID
Single-instance Oracle database	orlracpr
Policy-managed Oracle RAC instance	orlracpr_1
Admin-managed Oracle RAC instance	orlracpr1
Oracle RAC One Node database instance	orlracpr_1

2.9 Creating Database Passwords

Apply the following guidelines when specifying passwords:

- Passwords must be between 8 and 30 characters long.

- Passwords must be from the ASCII character set.

- Passwords must not start with a numeral.

- Passwords must not be the same as the user name.

- Passwords must not be Oracle reserved words.

- The SYS account password must not be `change_on_install`.

- The SYSTEM account password must not be `manager`.

- To use the same password for all the accounts, you must use a password that is not `change_on_install`, `manager`, or `dbsnmp`.

- Passwords should have at least 1 alphabetic, 1 numeric, and 1 punctuation character.

- Passwords should not be simple or obvious words, such as `welcome`, `account`, `database`, `oracle`, or `user`.

2.10 Understanding the Actions of OUI and DBCA During Installation

After installing Oracle Database, OUI runs DBCA to create your database, in accordance with Optimal Flexible Architecture (OFA) guidelines. An installation created following OFA guidelines means that DBCA creates your database files, including the default server parameter file (SPFILE), using standard file naming and file placement practices.

You can also use DBCA in standalone mode to create or delete a database.

> **See Also:**
>
> - Chapter 3, "Creating Oracle RAC or Oracle RAC One Node Databases with DBCA"
> - *Oracle Database Net Services Administrator's Guide* if you experience problems (for example, with the listener configuration), and for more information about Lightweight Directory Access Protocol (LDAP) support

2.10.1 Managing Database Services After Installation

You cannot use DBCA to manage database services for Oracle RAC databases. Use the Server Control Utility (SRVCTL), or use Oracle Enterprise Manager Database Express, or Oracle Enterprise Manager Cloud Control for all administration and monitoring of database services for an Oracle RAC database.

Oracle Enterprise Manager Cloud Control is available separately on the Oracle Enterprise Manager Cloud Control installation media, and on the Oracle Technology Network website at the following website:

http://www.oracle.com/technetwork/indexes/documentation/index.html

> **See Also:**
>
> *Oracle Enterprise Manager Real Application Clusters Guide Online Help* for service management using Oracle Enterprise Manager

2.11 Using Oracle Universal Installer to Install Oracle RAC

To install Oracle Database 12*c* Release 1 (12.1) with Oracle RAC:

1. Ensure that you can access other nodes with SSH. OUI requires that the user account running the Oracle RAC installation is permitted to set up passwordless SSH. OUI can set this up for you automatically, or your system administrator can set this up for you before installation is started.

2. Open the terminal from which you intend to run OUI, and log in as the user account that you want to own the Oracle Database installation (for example, `oracle`).

   If you are not able to turn off `stty` commands, or have other restrictions that would prevent automatic SSH configuration from within OUI, then you must ensure that SSH is configured and enabled before you proceed to start OUI.

3. Ensure that you have `su` or `sudo` credentials, because you will be prompted to run a script as the `root` user during installation.

4. Start the `runInstaller` command from the Oracle Database 12*c* Release 1 (12.1) installation media or install binary download area. For example, if the software binary download area is `/home/oracle/oraclesw`, enter the following commands:

   ```
   $ cd /home/oracle/oraclesw
   $ ./runInstaller
   ```

5. Provide information or run scripts as `root` when prompted by OUI. If you need assistance during installation, click **Help**. Click **Details** to examine the log file.

   > **Note:** You can run the `root.sh` script simultaneously on all nodes in the cluster for Oracle RAC installations or upgrades.

   If you encounter problems during installation, then examine the OUI actions recorded in the installation log file. The log file is located in the Oracle Inventory directory with a name that includes the time stamp (*date_time*) of the installation process, as shown in this example:

   OraInventory/logs/installActions*date_time*.log

   > **Caution:** Do not install the Oracle Database with Oracle RAC software into the same home in which you installed Oracle Grid Infrastructure software.
   >
   > Note that because Oracle ASM is part of the Oracle Grid Infrastructure home, it cannot be upgraded during an Oracle RAC installation.

When you have completed the second and final phase of the installation, proceed to Chapter 5, "Oracle Real Application Clusters Postinstallation Procedures" to perform the postinstallation tasks.

> **Caution:** After you complete the installation create the database, if you decide to install additional Oracle Database 12*c* Release 1 (12.1) products in the database, then you must stop *all* processes running in the Oracle home before installing the additional products. For OUI to relink certain executables and libraries, you must stop all database processes. Refer to Appendix D, "How to Stop Processes in an Existing Oracle RAC Database" for additional information.

2.12 Using Oracle Universal Installer to Install Oracle RAC One Node

To install Oracle RAC One Node:

1. Open the terminal from which you intend to run OUI, and log in as the user account that you want to own the Oracle Database installation (for example, `oracle`).

   If you are not able to turn off `stty` commands, or have other restrictions that would prevent automatic SSH configuration from within OUI, then you must ensure that SSH is configured and enabled before you start OUI.

2. Ensure that you have `su` or `sudo` credentials, because you will be prompted to run a script as the root user during installation.

3. Start the runInstaller command from the Oracle Database 12*c* Release 1 (12.1) installation media or install binary download area. For example, where the software binary download area is `/home/oracle/oraclesw`, enter the following commands:

   ```
   $ cd /home/oracle/oraclesw
   $ ./runInstaller
   ```

4. Select to configure an Oracle RAC One Node database, and select to install Oracle RAC on all nodes where Oracle RAC One Node will run.

5. Provide information when prompted by OUI. If you need assistance during installation, click **Help**. Click **Details** to see the log file.

> **Note:** You can run the `root.sh` script simultaneously on all nodes in the cluster for Oracle RAC installations or upgrades.

If you encounter problems during installation, then examine the OUI actions recorded in the installation log file. The log file is located in the Oracle Inventory directory with a name that includes the timestamp (*date_time*) of the install process, as shown in this example:

OraInventory/logs/installActions*date_time*.log

> **Caution:** The Oracle home name and path that you provide during database installation *must be different* from the home that you used during the Oracle Grid Infrastructure installation in phase one. You *must not* install Oracle Database with Oracle RAC software into the same home in which you installed the Oracle Grid Infrastructure for a cluster software.
>
> Note that because Oracle ASM is part of the Oracle Grid Infrastructure home, it cannot be upgraded during an Oracle RAC installation.

2.13 Installing the Oracle Database Vault Option

Installing and configuring Oracle Database Vault requires actions during and after installation, as described in the following topics:

- Starting the Listener with Oracle Database Vault Installations

- Configuring Oracle Database Vault Using DBCA

> **See Also:** *Oracle Database Vault Administrator's Guide* for information about the Database Vault accounts and roles that are configured during installation

2.13.1 Starting the Listener with Oracle Database Vault Installations

You must start the listener and database instance on all Oracle RAC nodes including the node on which the installation is performed. You must use Server Control (SRVCTL) to start and stop the Oracle RAC instances being configured for Oracle Database Vault. Do not use SQL*Plus to start and stop Oracle RAC instances.

2.13.2 Configuring Oracle Database Vault Using DBCA

You can configure Oracle Database Vault after installation using DBCA, or choose not to configure Oracle Database Vault.

To install Oracle Database Vault using DBCA:

1. After installing Oracle RAC, create the database.

2. Start DBCA and select the option **Configure Database**.

3. In the component list, select **Oracle Label Security** and **Oracle Database Vault**.

4. Provide the required Oracle Database Vault user IDs and passwords to proceed with configuration. To enable a separate Oracle Database Vault administrator, choose to configure the DV_ACCTMGR user.

5. After you have finished, restart each database instance to finish the software configuration.

> **See Also:** *Oracle Database Vault Administrator's Guide* for information about using Oracle Data Guard with Oracle Database Vault

3

Creating Oracle RAC or Oracle RAC One Node Databases with DBCA

This chapter describes how to use Database Configuration Assistant (DBCA) in standalone mode to create and delete Oracle Real Application Clusters (Oracle RAC) databases. The topics in this chapter include the following:

- Using DBCA with Oracle RAC or Oracle RAC One Node

- Benefits of Using Database Configuration Assistant

- Automatic Listener Migration from Earlier Releases

- Verifying Requirements for DBCA

- Creating an Oracle RAC or Oracle RAC One Node Database with DBCA

- Using DBCA to Create an Oracle RAC One Node Database

- Deleting an Oracle RAC Database Using DBCA

3.1 Using DBCA with Oracle RAC or Oracle RAC One Node

DBCA has the following primary database functions:

- Create and delete databases

- Create database templates

- Create, plug, unplug, and delete pluggable databases (PDBs)

- Add and delete database instances

- Register databases in Oracle Enterprise Manager Cloud Control

- Configure database options (such as Oracle Database Vault) and register database with the Directory Server

> **Note:** Cluster Managed Services are no longer managed through DBCA. Instead, use the Cluster Managed Services page in Oracle Enterprise Manager Cloud Control, if available, or SRVCTL. For more information, see *Oracle Real Application Clusters Administration and Deployment Guide*.

> **See Also:**
>
> - "Creating an Oracle RAC or Oracle RAC One Node Database with DBCA" on page 3-3 for more information about using DBCA in standalone mode
>
> - *Oracle Database Net Services Administrator's Guide* to resolve problems—for example, with the listener configuration—and for further information about Lightweight Directory Access Protocol (LDAP)-compliant directory support

3.2 Benefits of Using Database Configuration Assistant

Oracle recommends that you use DBCA to create your Oracle RAC database, because preconfigured databases optimize your environment for features such as the server parameter file (SPFILE), and automatic undo management.

DBCA enables you to create both policy-managed and administrator-managed databases. With DBCA, you can also create site-specific tablespaces as part of database creation. If you have data file requirements that differ from those offered by DBCA templates, then create your database with DBCA and modify the data files later. You can also run user-specified scripts as part of your database creation process.

DBCA also configures your Oracle RAC environment for various Oracle high availability features, such as cluster administration tools. DBCA also starts any database instances required to support your defined configuration.

You can use DBCA to create a database from templates supplied by Oracle, or from templates that you create. The templates contain settings optimized for a particular type of workload.

Oracle ships templates for the following two workload types:

- General purpose or transaction processing

- Data warehouse

For more complex environments, you can select the **Custom Database** option. This option does not use templates and results in a more extensive interview, which means that it takes longer to create your database.

Click **Show Details** to see the configuration for each type of database. Select the template suited to the type of workload your database will support. If you are not sure which to choose, then select the default **General Purpose or Transaction Processing** template.

3.3 Automatic Listener Migration from Earlier Releases

If your system has an Oracle Database 10*g* or 11*g* installation, and you install Oracle Database 12*c* Release 1 (12.1) either to coexist with or to upgrade the Oracle Database 10.1, 10.2, 11.1, or 11.2 installation, then most installation types automatically migrate the existing Oracle Database listener to the 12*c* Release 1 (12.1) Oracle home. During migration, the upgrade process configures and starts a default Oracle Net listener using the same TCP/IP port as the existing listener, with the IPC key value.

During the Oracle Clusterware upgrade, the default listener (LISTENER_*NODENAME*) was migrated to the Oracle Grid Infrastructure home (Grid home). DBCA always uses the default listener.

The listener migration process stops the listener in the existing Oracle home, and restarts it in the new Oracle home. If the database was using the default listener

(LISTENER_*NODENAME*), then Oracle Net Configuration Assistant (NETCA) migrates the listener automatically to the Grid home as part of the Oracle Grid Infrastructure upgrade. If the database was using a nondefault listener, then Database Upgrade Assistant (DBUA) migrates the nondefault listener to the Oracle Database home.

> **Note:**   During migration, client applications may not be able to connect to any databases that are registered to the listener that is being migrated.

3.4 Verifying Requirements for DBCA

To use Database Configuration Assistant (DBCA) to change the database configuration, run Cluster Verification Utility (CVU) to verify that your system is prepared for configuration changes using the following command syntax:

```
/Grid_home/bin/cluvfy stage -pre dbcfg -fixup -n node_list -d Oracle_home
[-verbose]
```

In the preceding syntax example, the variable `Grid_home` is the Oracle Grid Infrastructure home, the variable `node_list` is the list of nodes in your cluster, separated by commas, and the variable `Oracle_home` is the path for the Oracle home directory where OUI creates or modifies the database. The -fixup flag generates a fixup script that can be run as root to resolve many operating system configuration tasks if they were not completed before you run the check.

For example, to verify that your system is prepared for an Oracle Database with Oracle RAC installation on a two-node cluster with nodes node1 and node2, with the Oracle Grid Infrastructure home path/u01/app/12.1.0/grid, and with the Oracle home path /u01/app/oracle/product/12.1.0/db1, enter the following command:

```
$ /u01/app/12.1.0/grid/bin/cluvfy stage -pre dbcfg -fixup -n node1,node2 -d\
/u01/app/oracle/product/12.1.0/db1
```

You can select the option -verbose to receive progress updates as CVU performs its system checks, and detailed reporting of the test results.

If the CVUsummary indicates that the cluster verification check fails, and you cannot resolve these issues by running the fixup script, then review and correct the relevant system configuration steps, and run the test again.

3.5 Creating an Oracle RAC or Oracle RAC One Node Database with DBCA

To create a database with DBCA, you must have run the Oracle Net Configuration Assistant (NETCA) to configure your Oracle Net `listener.ora` file. You no longer need to set the operating system environment variables ORACLE_HOME to the Oracle RAC database home, or ORACLE_UNQNAME to the database unique name unless you are using Oracle Transparent Data Encryption with Oracle RAC.

> **Note:**   You can no longer set up email notification for Oracle RAC databases either from DBCA or OUI.

This section contains the following topics about using DBCA with Oracle RAC:

- [Tasks to Complete Before Using DBCA to Create any Oracle RAC Database](#)
- [Using DBCA to Create any Oracle RAC Database](#)
- [Using DBCA to Create an Oracle RAC One Node Database](#)

3.5.1 Tasks to Complete Before Using DBCA to Create any Oracle RAC Database

Before you can create an Oracle RAC database using DBCA you must configure your system to meet the software requirements, if this was not done as part of the Oracle Grid Infrastructure installation.

The following sections describe some common tasks you must perform:

- [Load SSH Keys Into Memory Before Starting DBCA](#)
- [Decide on a Naming Convention to Use for Your Oracle RAC Database](#)
- [Configure Shared Storage for the Oracle RAC Database](#)

3.5.1.1 Load SSH Keys Into Memory Before Starting DBCA

In an Oracle RAC environment, you must load SSH keys into memory for the terminal session where you start DBCA. If you do not do this, then you receive user equivalency errors when you attempt to start DBCA. If you use a pass phrase on your system for SSH, then you must provide the pass phrase to load the SSH keys.

Use the following commands to load SSH keys:

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

If needed, provide the pass phrase when prompted. You can then start DBCA.

3.5.1.2 Decide on a Naming Convention to Use for Your Oracle RAC Database

The global database name can be up to 30 characters in length, and must begin with an alphabetic character. The domain portion of the global database name can be no more than 128 characters and can contain only alphabetic and numeric characters, and the period (.) character.

The maximum number of characters you can use for the SID prefix is 8 characters. DBCA uses the SID prefix to generate a unique value for the variable ORACLE_SID for each instance. The SID prefix must begin with an alphabetic character.

3.5.1.3 Configure Shared Storage for the Oracle RAC Database

Before starting DBCA to configure an Oracle RAC database, you must have installed Oracle Grid Infrastructure for a cluster, and you must have configured shared storage areas for Oracle RAC files. Storage administration tasks require the SYSASM system privileges, which are granted to members of the OSASM operating system group. This group may not be the same as the OSDBA group, whose members are granted the SYSDBA system privileges.

See *Oracle Grid Infrastructure Installation Guide* for your platform for more information about shared storage configuration requirements.

3.5.2 Using DBCA to Create any Oracle RAC Database

This section describes how to use DBCA to create an Oracle RAC database or Oracle RAC One Node database:

3.5.2.1 Starting DBCA

To start DBCA, connect as the installation owner account (for example, `oracle`) to one of your nodes where Oracle RAC is installed, load SSH keys into memory, and enter the command `dbca` from the `$ORACLE_HOME/bin` directory.

> **Note:** You no longer need to set the operating system environment variables ORACLE_HOME to the Oracle RAC database home, or ORACLE_UNQNAME to the database unique name. However, if you are using Oracle Transparent Data Encryption with Oracle RAC, then you need to configure the ORACLE_UNQNAME parameter.

3.5.2.2 Cluster Detection and Node Selection when Using DBCA

When you start DBCA, it automatically shows Oracle RAC options if it detects from the central Oracle Inventory that the Oracle home is enabled for Oracle RAC.

If DBCA does not automatically display Oracle RAC options, then DBCA was unable to detect if the Oracle home is installed on a cluster. In this case, check that the OUI inventory is correctly located in the directory `/etc/oraInst.loc`, and that the `oraInventory` file is not corrupted. Also, perform clusterware diagnostics by using the following CVU command syntax:

`/Grid_home/bin/cluvfy/cluvfy.sh stage -post crsinst -n nodelist`

For example, with the mountpoint `/u01/app/12.1.0/grid`, and nodes `node1` and `node2`, run the following command:

`$ /u01/app/12.1.0/grid/bin/cluvfy stage -post crsinst -n node1,node2`

Note that when using DBCA, if nodes that are part of your cluster installation do not appear on the Node Selection page, then run the Opatch `lsinventory` command to perform inventory diagnostics. Also use CVU to perform clusterware diagnostics.

On the Management Options page, you are provided options for managing the database, either with Oracle Enterprise Manager Database Express or Oracle Enterprise Manager Cloud Control. For Oracle RAC databases, Oracle Enterprise Manager Database Express is configured to connect to the cluster using the Single Client Access Name (SCAN).

3.5.2.3 Using DBCA to Select Storage to Use With any Oracle RAC Database

You can choose to use either Oracle ASM diskgroups or a supported cluster file system as storage.

On the Specify Database Storage Options page, if you do not see the diskgroups in DBCA, then either Oracle ASM is not configured, or diskgroups are not mounted. You can create diskgroups using ASMCA in the Grid Infrastructure home before starting DBCA.

If you are using Oracle ASM or cluster file system storage, then you can also select the Fast Recovery Area and size on this page. If you are using Oracle ASM, then the Fast Recovery Area defaults to the Oracle ASM Disk Group.

3.5.2.4 Using DBCA to Specify Database Initialization Parameters for Oracle RAC

On the Initialization Parameters page, if you intend to add more nodes in your cluster than you have during the current DBCA session, then click **All Initialization Parameters**, and change the parameter CLUSTER_DATABASE_INSTANCES to the total number of nodes that you plan to add to the cluster.

In addition, if you click **All Initialization Parameters**, note that if your global database name is longer than 8 characters, then the database name value (in the DB_NAME parameter) is truncated to the first 8 characters, and the DB_UNIQUE_NAME parameter value is set to the global name.

If you are installing on a Linux system, note that **Memory Size (SGA and PGA)**, which sets the initialization parameter MEMORY_TARGET or MEMORY_MAX_TARGET, cannot be greater than the shared memory file system on your operating system.

For example, if the shared memory file system allocation on your system is 1 GB, but you set Memory Size (MEMORY_TARGET) to 2 GB, then you receive the following error during database startup:

```
ORA-00845: MEMORY_TARGET not supported on this system
ORA-01078: Failure in processing system parameters
```

This issue is not relevant for other platforms.

> **See Also:** *Oracle Database Administrator's Guide* for information about initialization parameters

3.5.2.5 Actions Performed By DBCA for Oracle RAC Databases

After you respond to DBCA prompts, review the Summary dialog information and click **OK**, DBCA does the following:

- Creates an Oracle RAC database, and its instances
- Creates the Oracle RAC data dictionary views
- Starts the CRS (Cluster Ready Service) resource for the database

> **Caution:** After you have created the Oracle RAC database, if you decide to install additional Oracle Database products in the Oracle RAC database you have created, then before you attempt to install the products, you must stop all processes running in the Oracle RAC database homes.
>
> You must stop all processes running in the Oracle RAC homes so that Oracle Universal Installer can relink certain executables and libraries. Refer to Appendix D, "How to Stop Processes in an Existing Oracle RAC Database" for additional information.

3.5.3 Using DBCA to Create an Oracle RAC One Node Database

If you have selected to install only the Oracle RAC software on cluster nodes, then you can use Oracle Database Configuration Assistant (DBCA) to configure Oracle RAC One Node.

After installation, start DBCA. From the Database Operation page, select the option **Create Database**. On the Creation Mode page, select **Advanced Mode**. On the Database Template page, select **Oracle RAC One Node database**.

Selecting one node deploys Oracle RAC One Node on a single node. Oracle recommends that you select all nodes in the cluster to which you want Oracle RAC One Node to be able to fail over.

If you select less than two nodes, or create server pools with a cardinality of 2 or more, then DBCA posts a warning message that the configuration you select will not support failover of the Oracle RAC One Node instance.

When you create an administrator-managed Oracle RAC One Node database, note that while the database is started on only one of the pool of nodes you installed the binaries, all the candidate servers are placed into the Generic server pool. If the servers are not already in Generic or Free, then this may result in stopping resources that are running on candidate servers.

When you use DBCA to create an Oracle RAC One Node database, a failover service is automatically configured.

3.6 Deleting an Oracle RAC Database Using DBCA

Deleting an Oracle RAC database using DBCA involves first deleting the database, and then removing the database's initialization parameter files, instances, Optimal Flexible Architecture (OFA) structure, and the Oracle network configuration for the database.

To delete a database using DBCA:

1. Start DBCA on one of the nodes:

   - Run the dbca command from the $ORACLE_HOME/bin directory.

   DBCA displays the Operations page, displaying different database deployment options.

2. Select **Delete a database**, and click **Next**. DBCA displays a list of all Oracle RAC and single-instance databases running from the Oracle home where DBCA is run.

3. If your user ID and password are not operating-system authenticated, then the List of Cluster Databases page displays the user name and password fields. If these fields appear, then enter a user ID and password for a user account that has SYSDBA privileges.

4. Select the database to delete, and click **Finish**.

   After you click **Finish**, DBCA displays a dialog box to confirm the database and instances that you have configured DBCA to delete.

5. Click **OK** to begin the deletion of the database and its associated files, services, and environment settings, or click **Cancel** to stop the operation.

When you click **OK**, DBCA continues the operation and deletes all the associated instances for this database. DBCA also removes the parameter files, password files, and oratab entries.

At this point, you have accomplished the following:

- Deleted the selected Oracle RAC database from the cluster

- Deleted high availability services assigned to the Oracle RAC database

- Deleted the Oracle Net configuration for the Oracle RAC database

- Deconfigured Oracle Enterprise Manager for the Oracle RAC database
- Deleted the OFA directory file structure for that Oracle RAC database from the cluster
- Deleted the Oracle RAC database data files

4

Deploying Oracle RAC Databases with Rapid Home Provisioning

This chapter describes how to deploy Oracle Real Application Clusters (Oracle RAC) databases using Rapid Home Provisioning and contains the following sections:

- Understanding Rapid Home Provisioning for Oracle RAC
- Using Rapid Home Provisioning to Deploy Oracle RAC Database Software
- Deploying Database Software and Creating Databases
- Patching Databases

4.1 Understanding Rapid Home Provisioning for Oracle RAC

Rapid Home Provisioning is a method of deploying software homes and databases to nodes in a cloud computing environment from a single cluster where you store templates of Oracle homes as images (called gold images) of Oracle software, such as databases, middleware, and applications. Using a working copy, which is a writeable Oracle Automatic Storage Management Cluster File System (Oracle ACFS) snapshot of the gold image, you can deploy an Oracle home, for example, to the Oracle Grid Infrastructure once and then use that home from any node in the cloud, without needing to provision the home each time you add a node to the cloud.

Gold images can be deployed in the same cluster as the Rapid Home Provisioning Server (RHPS) or a different cluster where a Rapid Home Provisioning Client (RHPC) is installed.

Table 4–1 indicates the various supported database releases and storage options for the database software for different releases:

Table 4–1   Supported Database Releases and Software Storage Options

Supported Release	Local Non-Shared	Local Shared	Network file system (NFS)	RHP_ MANAGED[1] (Oracle ACFS)
12.1.0.x	Yes	Yes	Yes	Yes
11.2.0.x	Yes	Yes	Yes	Yes
10.2.0.5	Yes	Yes	No	No

[1] Storage provided by the Rapid Home Provisioning infrastructure

4.2 Using Rapid Home Provisioning to Deploy Oracle RAC Database Software

To provision Oracle RAC databases using Rapid Home Provisioning Server, the Rapid Home Provisioning Server must be configured and running in the Oracle Grid Infrastructure home, either on the cluster where the Oracle RAC database needs to be provisioned, or on a remote cluster. In the latter case, the cluster where the Oracle RAC database is to be provisioned, should be configured and running the Rapid Home Provisioning Client daemon.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for information about how to configure Rapid Home Provisioning

- *Oracle Clusterware Administration and Deployment Guide* for information about `srvctl` commands and Rapid Home Provisioning

This section contains the following topics:

- Creating the Gold Image
- Provisioning Oracle Homes Using Database Software

4.2.1 Creating the Gold Image

To provision an Oracle RAC database, the database software must be available as a gold image in the Rapid Home Provisioning Server cluster. To import the gold image, perform the following steps:

1. On any node in the Rapid Home Provisioning Server or Rapid Home Provisioning Client, run the Oracle Universal Installer and complete a software-only install of Oracle Database software. Download and install any required patches from My Oracle Support.

2. On the same node, create a gold image as follows:

   ```
   $ rhpctl import image -image image_name -path path
   ```

where *image_name* is the name of the gold image created by this command, and `path` is the top-level directory path where the Oracle RAC database software was installed in step 1.

After the above steps are completed, the image is available for provisioning of Oracle RAC databases.

4.2.2 Provisioning Oracle Homes Using Database Software

After creating a gold image on the Rapid Home Provisioning Server cluster, you can use the gold images to provision working copies on the server cluster or on remote clusters that are configured with the Rapid Home Provisioning Client daemon.

You can allow Rapid Home Provisioning Server to provide the storage space for the Oracle Home software or specify shared or non-shared storage path for the Oracle Database software.

4.3 Deploying Database Software and Creating Databases

This section contains the following topics:

- Creating Databases Using Templates
- Use Cases for Provisioning Databases
- Specifying Passwords for Provisioned Databases
- Deleting Databases on Provisioned Homes

4.3.1 Creating Databases Using Templates

You can use templates to create new databases and duplicate existing databases. The information in templates includes database options, initialization parameters, and storage attributes (for data files, tablespaces, control files, and online redo log files).

> **See Also:** ■*Oracle Database 2 Day DBA* for more information about templates
>
> - Section 3.1, "Using DBCA with Oracle RAC or Oracle RAC One Node"

To specify a template for the database creation, use the '`-dbtemplate`' option. You can specify the template location in any of the following ways:

- **An absolute path and filename**: In this case, the template file should be available on all nodes in the target cluster.

  ```
  -dbtemplate /u01/dbtemplates/db12/general_template.dbt
  ```

- **An image name and a relative path in the form of image-name:relative-path:** This form is useful where the administrator might designate a single image as a holder of all database templates. In this case, the template is available on the Rapid Home Provisioning Server cluster, and is copied to the target cluster during the database creation.

  ```
  -dbtemplate TEMPLATE_IMAGE:db12/general_template.dbt
  ```

4.3.2 Use Cases for Provisioning Databases

This section explains how to use `rhpctl` commands to provision database software and create databases. For information about `rhpctl` command options, see Table 4–2.

This section contains the following topics:

- Provisioning Oracle Database Locally and Remotely
- Provisioning Policy-Managed Databases Using Gold Images
- Provisioning Oracle RAC One Node Database
- Provisioning Multitenant Container Database
- Provisioning Administrator-Managed Databases from Gold Images
- Provisioning Databases Using Templates
- Provisioning Database in an Existing Oracle Home
- Provisioning Database in Oracle Flex Cluster Environment

4.3.2.1 Provisioning Oracle Database Locally and Remotely

Use the `rhpctl add workingcopy` command to provision database software and Oracle Database. You can run this command on the Rapid Home Provisioning Server or on the Rapid Home Provisioning Client. If you run the command on the Rapid Home Provisioning Server, you can provision an Oracle RAC database either locally on the Rapid Home Provisioning Server, or remotely on any Rapid Home Provisioning Client. When you run the `rhpctl` command on the Rapid Home Provisioning Server, use the `-client` option to specify the remote cluster. Refer to Table 4–2 for default values for the `add workingcopy` options.

Example 4–1 and Example 4–2 show local and remote provisioning of databases.

Example 4–1   Provision Oracle Database Software Home Locally

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
```

Example 4–2   Provision Oracle Database Software Home Remotely

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-client payrollCluster -user payuser
```

When you run the command from Rapid Home Provisioning Server or Rapid Home Provisioning Client, it provisions Oracle Database software on the cluster for the user executing the command, with the default values for `-storagetype` and `-path`, if not specified, as explained in Table 4–2.

For more information about the options for this use case, see the context-sensitive help for `rhpctl add workingcopy -help REMOTEPROVISIONING` on the command line.

4.3.2.2 Provisioning Policy-Managed Databases Using Gold Images

To provision Oracle Database software and create a policy-managed database from a gold image on an existing server pool, specify the server pool name using the `-serverpool` option. You can also provision Oracle Database software and create a policy-managed database on a new server pool using the `-newpool` option. In this case, specify the `-cardinality` option with a number to indicate the number of instances to be configured and started for this policy-managed database. The allocation of servers to this new pool depends on the current server and server pool configuration on the target cluster and CRS policies. Ensure that the Free server pool contains sufficient number of servers, otherwise other databases running on the nodes may be stopped to accommodate this database. The new pool is created for the Oracle Grid Infrastructure user (`grid`) with the minimum and maximum size set to the specified cardinality, and uses the default values for all other parameters.

Example 4–3, Example 4–4, and Example 4–5 show how to create policy-managed databases.

Example 4–3   Create a Policy-Managed Database from a Gold Image on an Existing Server Pool

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-dbname proddb -datafileDestination PAYDG1 -serverpool proddbpool1
```

Example 4–4   Create Policy-Managed Database from a Gold Image on a New Server Pool

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
```

```
-dbname proddb - datafileDestination PAYDG1
-newpool proddbpool1 -cardinality 4
```

Example 4–5   Create a Policy-Managed Database in LOCAL or User-Specified Location

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-storagetype LOCAL -path /u01/app/product/12.1.0/dbhome -dbname proddb
-datafileDestination PAYDG1
-newpool proddbpool1 -cardinality 4
```

When you run the command from Rapid Home Provisioning Server or Rapid Home Provisioning Client, it provisions Oracle Database software on the cluster for the user executing the command with the default values for `-storagetype` and `-path`, if not specified, as explained in Table 4–2. It also creates a policy-managed Oracle RAC database on the specified server pool with database files located in the `-datafileDestination` path.

For more information about the options for this use case, see the context-sensitive help for `rhpctl add workingcopy -help POLICYDB` and `rhpctl add workingcopy -help STORAGETYPE` on the command line.

4.3.2.3 Provisioning Oracle RAC One Node Database

To specify the type of database to provision, use the `-dbtype` option. For example, to provision Oracle Database software and create a policy-managed Oracle RAC One Node database from a gold image:

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-dbtype RACONENODE -dbname proddb -datafileDestination PAYDG1
-serverpool proddbpool1
```

4.3.2.4 Provisioning Multitenant Container Database

To provision Oracle Database software and create a policy-managed multitenant container database (CDB), specify the server pool name, the database files location, and the name and number of pluggable databases (PDBs):

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-dbname proddb -datafileDestination PAYDG1
-serverpool proddbpool1
-cdb -pdbName prodpdb -numberOfPDBs 10
```

When you run the command from Rapid Home Provisioning Server or Rapid Home Provisioning Client, it provisions Oracle Database software on the cluster for the user executing the command with the default values for `-storagetype` and `-path`, if not specified, as explained in Table 4–2. It also creates a policy-managed Oracle RAC multitenant container database (CDB) on the server pool and ten PDBs named `prodpdb1`, `prodpdb2`, and so on. The database files are located in the `-datafileDestination` path.

For more information about the options for this use case, see the context-sensitive help for `rhpctl add workingcopy -help PDB` on the command line.

4.3.2.5 Provisioning Administrator-Managed Databases from Gold Images

To provision Oracle Database software and create an administrator-managed database from a gold image, specify the list of nodes where the Oracle RAC database instances

need to be configured and started, using the `-node` option. You must specify existing nodes on the target cluster in the list of nodes. Ensure that the nodes are in the Free or Generic server pool on the target cluster. If you specify nodes that are not in the Free or Generic pool, then policy-managed databases running on the nodes may be stopped to accommodate this database. Specify the type of database to provision using the `-dbtype` option.

Example 4–6, Example 4–7, and Example 4–8 show how to create administrator-managed databases.

Example 4–6   Create a Administrator-Managed Oracle RAC Database from a Gold Image

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-dbname proddb -datafileDestination PAYDG1 -node prodsrvr1,prodsrvr2,prodsrvr3
```

Example 4–7   Create a Administrator-Managed Oracle RAC One Node Database from a Gold Image

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-dbtype RACONENODE -dbname proddb -datafileDestination PAYDG1 -node
prodsrvr1,prodsrvr2,prodsrvr3
```

Example 4–8   Create a Single-Instance Oracle Database from a Gold Image

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-dbtype SINGLE -dbname proddb -datafileDestination PAYDG1 -node prodsrvr1
```

When you run the command from Rapid Home Provisioning Server or Rapid Home Provisioning Client, Oracle Database software is provisioned on the cluster for the user executing the command with the default values for `-storagetype` and `-path`, if not specified, as explained in Table 4–2. It also creates administrator-managed databases, single-instance, Oracle RAC, or Oracle RAC One Node, as per the `-dbtype` option on the specified nodes, with database files located in the `-datafileDestination` path.

For more information about the options for this use case, see the context-sensitive help for `rhpctl add workingcopy -help ADMINDB` on the command line.

4.3.2.6 Provisioning Databases Using Templates

To provision Oracle Database software and create a database from a database template image, specify the template location using the `-dbtemplate` option with the absolute file path of the template file or the relative path to the image home directory on the Rapid Home Provisioning Server. To specify an absolute file path for the template, the template must exist in the target cluster, on all nodes for a policy-managed database and on the nodes you specify in the command-line for administrator-managed database.

Example 4–9 and Example 4–10 show how to provision databases using templates.

Example 4–9   Create a Policy-Managed Database using Rapid Home Provisioning Server (RHPS) Image

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-dbname proddb -datafileDestination PAYDG1 -serverpool proddbpool1
 -dbtemplate DBTMPLTIMG:templates/database/12.1.0/warehouse.dbt
```

Example 4–10 Create a Policy-Managed Database using Template Stored in a User-Defined Location

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12  -oraclebase
/u01/app/product/12.1.0/obase
-dbname proddb  -datafileDestination PAYDG1 -serverpool proddbpool1
-dbtemplate /templates/database/12.1.0/warehouse.dbt
```

When you run the command from Rapid Home Provisioning Server or Rapid Home Provisioning Client, it provisions Oracle Database software on the cluster for the user executing the command with the default values for `-storagetype` and `-path`, if not specified, as explained in Table 4–2. It also creates a database on the specified server pool from the database template.

For more information about the options for this use case, see the context-sensitive help for `rhpctl add workingcopy -help DBTEMPLATE` on the command line.

4.3.2.7 Provisioning Database in an Existing Oracle Home

Use the `rhpctl add database` command to provision a new database in an existing provisioned database software home and server pool. For example:

```
$ rhpctl add database -workingcopy mydbhome -dbname ORCL2
-datafileDestination PAYDG1 -serverpool pool2
```

The command creates a new database ORCL2 and starts it on all nodes that are active in the server pool pool2.

4.3.2.8 Provisioning Database in Oracle Flex Cluster Environment

Provision Oracle Database software and create a database in an existing server pool and pqpool in Oracle Flex Cluster environment:

```
rhpctl add workingcopy -workingcopy myDBHome1 -image ORACLEDB12 -oraclebase
/u01/app/product/12.1.0/obase
-dbname proddb  -dbtype RAC -datafileDestination PAYDG1 -serverpool proddbpool1
-pqpool proddbpool2
```

For more information about the options for this use case, see the context-sensitive help for `rhpctl add workingcopy -help DBWITHPQPOOLS` on the command line.

Table 4–2 lists the parameters for the `rhpctl` command.

Table 4–2 rhpctl Parameters for Provisioning Databases

Option	Description
`-workingcopy workingcopy_name`	Working copy to provision the database
`-image image_name`	Image name
`-oraclebase oraclebase_path`	Oracle base location for provisioning Oracle Database home
`-dbname unique_db_name`	Unique name for the database
`-datafileDestination datafile_destination_path`	Path to store database data files or Oracle ASM disk group name.
`-serverpool pool_name`	Server pool in which the policy-managed database is provisioned
`-node node_list`	A node or comma-delimited list of several nodes on which to create the administrator-managed database

Table 4–2   (Cont.) rhpctl Parameters for Provisioning Databases

Option	Description
`-pdbName pdb_prefix`	The PDB name prefix if one or more PDBs need to be created.
`-numberOfPDBs pdb_count`	Number of PDBs
`-dbTemplate file_path` or `-dbTemplate image-name:relative_file_path`	Absolute file path for the template file or the relative path to the image home directory on Rapid Home Provisioning Server. If this option is not specified on the command line, then a new General Purpose Database is created.
`-storagetype {NFS|LOCAL|RHP_MANAGED}`	Type of storage used for the provisioned Oracle home. The possible values are: ■ `RHP_MANAGED`: Storage provided by the Rapid Home Provisioning infrastructure, that is Oracle ACFS. ■ `LOCAL`: Storage that you specify on a file system on the target cluster. ■ `NFS`: Storage provided by Rapid Home Provisioning infrastructure on the server cluster and NFS-mounted on the target cluster. If both `-storagetype` and `-path` options are not specified, then Rapid Home Provisioning will attempt to use `RHP_MANAGED` storage.
`-path storage_path`	Path where Oracle Database software is provisioned when using the LOCAL/NFS storage type. If the `-path` option is not specified, then there are two possibilities: ■ If Rapid Home Provisioning Client is configured with a disk group, then Oracle Database software is provisioned on `RHP_MANAGED` storage, that is, Oracle ACFS. ■ If Rapid Home Provisioning Client is not configured with a disk group, then a default Oracle home path, such as `oracle_base_path/product/db_version/workingcopy_name`, based on the value of the `-oraclebase` option is used to provision the database software. This path could be shared or non-shared. **Note:** To determine whether the Rapid Home Provisioning Client is configured with a disk group, run the `srvctl config rhpclient` command on the target cluster.
`-client cluster_name`	Name of the client cluster. If this option is not specified on the command line, then the provisioning operation is performed on the cluster from where the `rhpctl` command is run.
`-user user_name`	Owner of the provisioned Oracle Database software. If this option is not specified on the command line, then the provisioning operation is performed for the user executing the `rhpctl` command. When provisioning remotely, use the `-client` option to specify the client cluster and the `-user` option to specify the user on the client cluster.
`-dbtype {RACONENODE|RAC|SINGLE}`	Type of database to provision on the local Rapid Home Provisioning Server or remote Rapid Home Provisioning Clients. This can be `RAC`, `RACONENODE`, or `SINGLE`. If the `-dbtype` option is not specified on the command line, then an Oracle RAC database is created.

Table 4–2   (Cont.) rhpctl Parameters for Provisioning Databases

Option	Description
`-pqpool` *server_pool_name*	Name of an existing server pool.
	Note: This parameter is only applicable in an Oracle Flex Cluster environment and refers to server pools (either already defined as in this case, or to be created when you use the `-newpqpool` parameter) running on Leaf Nodes. If you create a new server pool, then you must specify a cardinality value for the server pool using the `-pqcardinality` parameter.

> **See Also:** ■*Oracle Clusterware Administration and Deployment Guide* for information about `rhpctl` commands for Rapid Home Provisioning
>
> ■ *Oracle Clusterware Administration and Deployment Guide* for information about server allocation to server pools

4.3.3 Specifying Passwords for Provisioned Databases

There is no mechanism to specify passwords for the schemas of a provisioned database. If the password needs to be known, it is recommended that the DBA user connects to the database using OS authentication on the node where the database instance is running, and reset the password to the desired value for each schema.

4.3.4 Deleting Databases on Provisioned Homes

Use the command `rhpctl delete database` to remove an existing database from a provisioned database software home. To ensure successful deletion, the database should not be running.

4.4 Patching Databases

You can patch databases in one of the following ways:

■   Patching the software

■   Patching either the data dictionary or schemas in the database, or both

See the following sections for more information:

■   Creating an Image for Database Patching

■   Use Cases for Patching

4.4.1 Creating an Image for Database Patching

Patching database software requires a one-time creation of a new image that includes the patched software. Follow these steps to create a database image:

1.   Create a working copy of an existing database image that needs to be patched.

2.   On the provisioned `ORACLE_HOME` which is the working copy, apply the necessary set of patches using OPatch or other mechanism.

3.   Run the command `rhpctl add image` to create a new image from the updated working copy. This image is the patched version of the original image.

    ```
    $ rhpctl add image -image image_name -workingcopy workingcopy_name
    ```

4. To include the image as part of an existing image series:

```
$ rhpctl add image -image image_name -workingcopy workingcopy_name -series
series_name
```

4.4.2 Use Cases for Patching

To patch a database, run the command `rhpctl move database` which creates a new `ORACLE_HOME` based on a patched image, if it does not exist, and also makes the database run from this new location instead of its current `ORACLE_HOME` location.

By default, patching is performed in rolling mode. Use the `-nonrolling` option to perform patching in non-rolling mode. The database is then completely stopped on the old `ORACLE_HOME`, and then restarted to make it run from the newly patched `ORACLE_HOME`.

For databases versions 12.1.0.1 or higher, the command `rhpctl move database` also executes SQL commands required for database patching. For database versions earlier than 12.1, a message is displayed asking the user to run the SQL commands for database patching manually.

This section contains the following use cases for patching:

- Moving Databases from Source Working Copy to Target Working Copy
- Creating a Working Copy Using a Database Image
- Moving Databases to an Oracle Home Managed by Rapid Home Provisioning Server

4.4.2.1 Moving Databases from Source Working Copy to Target Working Copy

To move all databases running from a source working copy to a target working copy in rolling mode:

```
rhpctl move database -sourcewc myDBHome1 -patchedwc myDBHome2
```

To move a specific database running from a source working copy to a target working copy in rolling mode:

```
rhpctl move database -sourcewc myDBHome1 -patchedwc myDBHome2 -dbname proddb
```

When you run the command from Rapid Home Provisioning Server or Rapid Home Provisioning Client, it moves one or all databases from the cluster and Oracle home in the source working copy to the Oracle home in the target working copy in rolling mode.

4.4.2.2 Creating a Working Copy Using a Database Image

You can also specify an image to create a working copy. For example, to create a target working copy and move all databases running from the source working copy to the target working copy in rolling mode:

```
rhpctl move database -sourcewc myDBHome1 -patchedwc myDBHome2 -image ORACLEDB12
```

When you run the command from Rapid Home Provisioning Server or Rapid Home Provisioning Client, it creates a target working copy from the values specified in the image, and moves all databases running on the cluster from the Oracle home in the source working copy to the Oracle home in the target working copy in rolling mode. The storage type of the newly-created target Oracle home is the same as the source Oracle home and the storage path has the default value mentioned for the `-path`

option in Table 4–2. For a Rapid Home Provisioning Client operation, the default storage type is LOCAL.

> **Note:** You can move a source working copy from a non-shared Oracle home location to a target working copy in a shared Oracle home, but you cannot move a source working copy from a shared Oracle home location to a target working copy in a non-shared Oracle home.

4.4.2.3 Moving Databases to an Oracle Home Managed by Rapid Home Provisioning Server

You can move databases from an Oracle home that was not provisioned using Rapid Home Provisioning (unmanaged Oracle home) to an Oracle home provisioned and managed by Rapid Home Provisioning Server. Run the following command to bring Oracle Databases within the purview of the Rapid Home Provisioning server:

```
rhpctl move database -sourcehome /u01/app/product/12.1.0/dbhome -oraclebase
/u01/app/product/12.1.0/obase
-client payrollCluster -patchedwc myDBHome1 -image ORACLEDB12
```

When you run the command from Rapid Home Provisioning Server, it creates a new working copy myDBHome1 from the database software gold image ORACLEDB12 and moves all databases currently configured or running in the Oracle home /u01/app/product/12.1.0/dbhome to the new Oracle home.

Table 4–3 lists the parameters for the rhpctl commands for patching databases.

Table 4–3   rhpctl Parameters for Patching Databases

Option	Description
-sourcewc *oracle_home_ path*	Name of the old working copy from which the database needs to be moved
-patchedwc *oracle_home_ path*	Name of the new working copy to which the database needs to be moved
-image *image_name*	Name of the image
-dbname *unique_db_name*	Name of the database to be provisioned
-oraclebase *oraclebase_ path*	Oracle base path for provisioning Oracle database home (Required only for ORACLEDBSOFTWARE image type)
-client *client_cluster_ name*	Name of the client cluster
-sourcehome *oracle_home_ path*	Source Oracle home path

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for information about rhpctl commands for Rapid Home Provisioning

5

Oracle Real Application Clusters Postinstallation Procedures

This chapter describes how to complete the postinstallation tasks after you have installed the Oracle Database 12*c* Release 1 (12.1) with Oracle Real Application Clusters (Oracle RAC) software. This chapter contains the following topics:

- Required Postinstallation Tasks
- Recommended Postinstallation Tasks
- Oracle Configuration Manager Postinstallation Configuration for Oracle RAC
- Enabling and Disabling Database Options After Installation

> **Note:** This chapter describes only basic configurations. Refer to *Oracle Database Administrator's Guide*, and the product administration and tuning guides for more detailed configuration and tuning information. Refer also to *Oracle Database Installation Guide* for your platform for additional postinstallation configuration information.

5.1 Required Postinstallation Tasks

Perform the following tasks after completing your installation:

- Download and Install Software Patch Updates
- Set External Jobs Ownership for Installations on HP-UX
- Configure Oracle Products
- Recompile all PL/SQL modules
- Configuring Services on Oracle RAC and Oracle RAC One Node CDBs

5.1.1 Download and Install Software Patch Updates

Download and install patch updates. See the My Oracle Support website for required patch updates for your installation.

To download required patch updates:

1. Use a Web browser to view the My Oracle Support website:

   https://support.oracle.com

2. Log in to My Oracle Support.

> **Note:** If you are not a My Oracle Support registered user, then click **Register for My Oracle Support** and register.

3. On the main My Oracle Support page, click **Patches & Updates**.

4. On the Patches and Updates page, click **Product or Family (Advanced)**.

5. In the Product field, select **Oracle Database**.

6. In the Release field, select one or more release numbers. For example, Oracle 12.1.0.1.0.

7. Click **Search**.

8. Any available patch updates are displayed in the **Patch Search** page.

9. Click the patch number to download the patch.

10. Select the patch number and click **Read Me**. The README page contains information about the patch set and how to apply the patches to your installation.

11. Return to the Patch Set page, click **Download**, and save the file on your system.

12. Use the unzip utility provided with Oracle Database 12*c* Release 1 (12.1) to uncompress the Oracle patch updates that you downloaded from My Oracle Support. The unzip utility is located in the `$ORACLE_HOME/bin` directory.

13. See *Oracle Grid Infrastructure Installation Guide* for information about how to stop database processes in preparation for installing patches.

5.1.2 Set External Jobs Ownership for Installations on HP-UX

On HP-UX platforms only, complete the following procedure to set external jobs ownership to the low-privilege user `extjob`:

1. Log on as `root`.

2. Change directory to the Oracle Database Oracle home:

   ```
   # cd $ORACLE_HOME/rdbms/admin/
   ```

3. Open `externaljob.ora` with a text editor, and find the parameters `run_user` and `run_group`.

4. Set `run_user` to the external jobs user (`extjob`), and set `run_group` to a low-privileged group, such as `other`. For example:

   ```
   run_user=extproc
   run_group=other
   ```

5. Save the file.

> **Note:** Modify `externaljob.ora` only as `root`.

5.1.3 Configure Oracle Products

Many Oracle products and options must be configured before you use them for the first time. Before using individual Oracle Database 12*c* Release 1 (12.1) database products or options, refer to the manual in the product documentation library, which is available on the Oracle Technology Network (OTN) at:

http://docs.oracle.com

Refer to these sections for information about configuring various products and features after installation:

- Configuring Oracle Database Vault
- Configuring Oracle Database Security Settings
- Configuring Oracle Label Security
- Configuring Oracle XML DB
- Configuring Storage for External Tables, Shared Files, or Directory Objects

5.1.3.1 Configuring Oracle Database Vault

OUI installs Oracle Database Vault by default when you install the Oracle RAC software, but you must register Oracle Database Vault with the Oracle RAC database and create database user accounts before you can use it. You must create the Database Vault Owner user and, optionally, the Database Vault Account Manager administrative user accounts.

5.1.3.2 Configuring Oracle Database Security Settings

To enable or disable the database security configuration after installation, you must use command-line Oracle Database Configuration Assistant (DBCA) options. By design, the DBCA graphical user interface (GUI) does not have the option to enable or disable secure configuration. For example, to enable the security settings after installation, you would use a command of the following form, where myRACdb1.example.com is the either the name of the local database instance or the DB_ UNIQUE_NAME:

```
dbca -configureDatabase -sourceDB myRACdb1.example.com -SID
-enableSecurityConfiguration true
```

5.1.3.3 Configuring Oracle Label Security

After installation, you must configure Oracle Label Security in a database before you use it. You can configure Oracle Label Security in two ways: with Oracle Internet Directory integration and without Oracle Internet Directory integration.

Configuration	Requirement
With Oracle Internet Directory integration	To configure Oracle Label Security with Oracle Internet Directory integration, Oracle Internet Directory must be installed in your environment and the Oracle database must be registered in the directory.
Without Oracle Internet Directory integration	If you configure Oracle Label Security (OLS) without Oracle Internet Directory integration, then you cannot configure it to use Oracle Internet Directory at a later stage. To configure Oracle Label Security with Oracle Internet Directory on your database at a later time, you must remove the OLS option on the database and configure the OLS with Oracle Internet Directory integration option.

> **See Also:** *Oracle Label Security Administrator's Guide* for information about configuring Oracle Label Security

5.1.3.4 Configuring Oracle XML DB

Oracle XML DB is a required component of the Oracle Database installation. However, you must manually configure the FTP and HTTP ports for Oracle XML DB.

> **See Also:** *Oracle XML DB Developer's Guide* for information on configuring the FTP and HTTP protocols for Oracle XML DB

5.1.3.5 Configuring Storage for External Tables, Shared Files, or Directory Objects

If your Oracle RAC database uses files that are external to the database, then the external files should be located on shared storage that is accessible to all nodes. Each node should use the same mount point to access the file. Acceptable shared file systems include Database File System (DBFS), Oracle ASM Cluster File System (Oracle ACFS), or a supported network file system (NFS) using the Direct NFS Client.

The database directory object used to write and read files external to the database must point to a shared storage location, and each node must use the same mount point for the same shared storage location.

> **Note:** There is no checking of the contents of the external files or directory object specified as part of the external table to ensure that the directory contents are consistent on each node. To avoid unpredictable results, you must ensure that the same file is accessed from all nodes, or that the same file is used on all nodes.

> **See Also:** *Oracle Grid Infrastructure Installation Guide* for more information about configuring storage

5.1.4 Recompile all PL/SQL modules

Run the `utlrp.sql` script after creating or upgrading a database. This script recompiles all PL/SQL modules that might be in an invalid state, including packages, procedures, and types. You must do it immediately following installation, not at a later date.

1. Complete setting up the Oracle software owner user account environment (for example, `oracle`), as described in Section 5.2.1, "Set the Oracle User Environment Variables".

2. Start SQL*Plus, as follows:

   ```
   $ sqlplus "/ AS SYSDBA"
   ```

3. Run the `utlrp.sql` script, where `Oracle_home` is the Oracle home path:

   ```
   SQL> @Oracle_home/rdbms/admin/utlrp.sql
   ```

5.1.5 Configuring Services on Oracle RAC and Oracle RAC One Node CDBs

During installation, if you select a multitenant container database (CDB), and configure pluggable databases (PDBs), then you must add services to the PDBs after installation. If you do not add services to PDBs, and then the Oracle RAC One Node CDB fails over to another node, or you manually relocate the CDB to another node, then by default, all PDBs associated with the CDB that do not have registered services are restarted in `MOUNTED` state.

PDBs are opened in `Read Write` mode after failover, in case of Oracle RAC One Node database, or relocation in case of any Oracle RAC database, only after you have configured the PDBs to have associated services. If you have not associated services to PDBs, then the PDBs remains in `MOUNTED` state when the CDB instance restarts.

To add services to a PDB, use the following `srvctl` command syntax, where *cdbname* is the name of the CDB, *service_name* is the name of the service, and *pdbname* is the name of the PDB:

```
srvctl add service -d cdbname -s service_name -pdb pdbname
```

After you add services to your PDBs, if you relocate the CDB with which the PDBs are associated, or the CDB fails over for Oracle RAC One Node databases, then the PDBs associated with that CDB automatically open in `Read Write` state.

> **See Also:** *Oracle Database Concepts* and *Oracle Real Application Clusters Administration and Deployment Guide* for more information about PDBs and adding services

5.2 Recommended Postinstallation Tasks

Oracle recommends that you complete the following tasks after installing Oracle RAC:

- Set the Oracle User Environment Variables
- Set Up Additional User Accounts
- Downloading and Installing the ORAchk Health Check Tool

5.2.1 Set the Oracle User Environment Variables

On each node, in the installation owner user profile file (in this example, `oracle`), set the environment variables `ORACLE_BASE` and `ORACLE_HOME`, and `ORACLE_SID`; also add `ORACLE_HOME/bin` to the path environment variable.

For example:

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/12.1.0/dbhome_1
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID=sales1
```

If the environment variables `ORACLE_HOME` and `ORACLE_SID` are not set, and you try to use SQL*Plus or other tools, then you receive an error message requesting that you set these variables.

5.2.2 Set Up Additional User Accounts

You can set up additional user accounts to manage your database. For information about setting up additional optional user accounts, see *Oracle Database Security Guide*.

5.2.3 Downloading and Installing the ORAchk Health Check Tool

Download and install the ORAchk utility to perform proactive heath checks for the Oracle software stack.

ORAchk is an expansion of the RACCheck utility. ORAchk extends health check coverage to the entire Oracle software stack, and identifies and addresses top issues

reported by Oracle users. ORAchk proactively scans for known problems with Oracle products and deployments, including the following:

- Standalone Oracle Database

- Oracle Grid Infrastructure

- Oracle Real Application Clusters

- Maximum Availability Architecture (MAA) Validation

- Upgrade Readiness Validations

- Oracle Golden Gate

For information about configuring and running the ORAchk utility, refer to My Oracle Support note 1268927.1:

https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1268927.1

5.3 Oracle Configuration Manager Postinstallation Configuration for Oracle RAC

If you have installed Oracle Configuration Manager, then you must run a script to create a database account to collect database configuration collections. You must create this account in both **Connected** and **Disconnected** modes. The database account stores the PL/SQL procedures that collect the configuration information, and the account owns the database management system (DBMS) job that performs the collection. After the account has been set up, the account is locked because login privileges are no longer required.

To configure the database for configuration collection, run the following script:

```
$ORACLE_HOME/ccr/admin/scripts/installCCRSQL.sh collectconfig -s SID -r\
SYSDBA-USER -p SYSDBA-PASSWORD
```

The script `installCCRSQL.sh` creates an Oracle Configuration Manager user and loads the PL/SQL procedure into the database defined by the `ORACLE_SID`. You can also specify the database SID by using the `-s` option in the command line, as in the following example, where the SID is `orcl`:

```
$ORACLE_HOME/ccr/admin/scripts/installCCRSQL.sh collectconfig -s orcl
```

For Oracle RAC, you must run the database script against only one instance, such as the local instance on which you performed the installation. However, Oracle Configuration Manager must be installed in all instance homes.

> **See Also:** *Oracle Configuration Manager Installation and Administration Guide* for more information about Oracle Configuration Manager installation

5.4 Enabling and Disabling Database Options After Installation

When you install Oracle Database, some options are enabled and others are disabled. To enable or disable a particular database feature for an Oracle home, shut down the database and use the `chopt` tool.

The `chopt` tool is a command-line utility that is located in the directory `ORACLE_HOME/bin`. It uses the following syntax, where *db_option* is the option whose status you want to modify:

```
chopt [enable | disable] db_option
```

Table 5–1 shows the possible options for *db_option*:

Table 5–1   Chopt Tool Command Options for Database Options

Value	Description
dm	Oracle Data Mining Database Files
olap	Oracle OLAP
partitioning	Oracle Partitioning
rat	Oracle Real Application Testing

Example 5–1   How to Enable a Database Option Using Chopt

In this example, the chopt tool command enables Oracle Label Security in the database oradb1:

```
$ cd $ORACLE_HOME/bin
$ srvctl stop database -db oradb1
$ chopt enable lbac
$ srvctl start database -db oradb1
```

6

Using Server Pools with Oracle RAC

This chapter describes server pool concepts in Oracle Real Application Clusters (Oracle RAC) environments. This chapter contains the following topics:

- Policy-Managed Clusters and Capacity Management
- Oracle RAC Database and Server Pools
- About Creating Server Pools for Oracle RAC Databases
- Oracle RAC One Node and Server Pools

6.1 Policy-Managed Clusters and Capacity Management

Oracle Clusterware 11g release 2 (11.2) introduced server pools, where resources that Oracle Clusterware manages are contained in logical groups of servers called **server pools**. Resources are hosted on a shared infrastructure and are contained within server pools. Resources are no longer defined as belonging to a specific instance or node. Instead, the priority of resource requirements is defined.

In an Oracle Flex Cluster, with Hub Nodes and Leaf Nodes, you can use server pools to run particular types of workloads on cluster member nodes, while providing simplified administration options. You can use a cluster configuration policy set to provide dynamic management of cluster policies across the cluster.

The following sections provide additional concepts about server pools:

- Server Pools and Server Categorization
- Server Pools and Policy-Based Management
- How Server Pools Work
- Default Server Pools

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide*

6.1.1 Server Pools and Server Categorization

You can manage servers dynamically using server pools by identifying servers distinguished by particular attributes, a process called **server categorization**. In this way, you can manage clusters made up of heterogeneous nodes.

6.1.2 Server Pools and Policy-Based Management

With policy-based management, database administrators specify the server pool (excluding Generic or Free) in which the database resource runs

Policy-based management:

- Enables dynamic capacity assignment when needed to provide server capacity in accordance with the priorities you set with policies

- Enables allocation of resources by importance, so that applications obtain the required minimum resources, whenever possible, and so that lower priority applications do not take resources from more important applications

- Ensures isolation where necessary, so that you can provide dedicated servers in a cluster for applications and databases

- Enables policies to be configured to change pools in accordance with business needs or application demand, so that pools provide the right service at the right time

Applications and databases running in server pools do not share resources. Because server pools do not share resources, they isolate resources where necessary, but enable dynamic capacity assignments as required. Together with role-separated management, this capability addresses the needs of organizations that have standardized cluster environments, but allow multiple administrator groups to share the common cluster infrastructure.

Oracle Clusterware efficiently allocates different resources in the cluster. You need only to provide the minimum and maximum number of nodes on which a resource can run, combined with a level of importance for each resource that is running on these nodes.

> **See Also:**
>
> - *Oracle Clusterware Administration and Deployment Guide* for more information about resource attributes
>
> - *Oracle Clusterware Administration and Deployment Guide* for details about managing server pools to respond to business or application demand

6.1.3 How Server Pools Work

Server pools divide the cluster into groups of servers hosting singleton and uniform database services and applications. They distribute a uniform workload (a set of Oracle Clusterware resources) over several servers in the cluster. For example, you can restrict Oracle databases to run only in certain server pools. When you enable role-separated management, you can grant permission to operating system users to use server pools.

You manage server pools that contain Oracle RAC databases with the Server Control (SRVCTL) utility. Use the Oracle Clusterware Control (CRSCTL) utility to manage all other server pools. Only cluster administrators have permission to create top-level server pools.

Top-level server pools:

- Logically divide the cluster

- Are always exclusive, meaning that one server can only reside in one particular server pool at a certain point in time

6.1.4 Default Server Pools

When Oracle Clusterware is installed, two server pools are created automatically: *Generic* and *Free*. All servers in a new installation are assigned to the Free server pool, initially. Servers move from Free to newly defined server pools automatically.

6.1.4.1 The Free Server Pool

The Free server pool contains servers that are not assigned to any other server pools. The attributes of the Free server pool are restricted, as follows:

- `SERVER_NAMES`, `MIN_SIZE`, and `MAX_SIZE` cannot be edited by the user

- `IMPORTANCE` and `ACL` can be edited by the user

6.1.4.2 The Generic Server Pool

The Generic server pool stores any Oracle Database that is not policy managed. Additionally, the Generic server pool contains servers with names you specified in the `SERVER_NAMES` attribute of the server pools that list the Generic server pool as a parent server pool.

The Generic server pool's attributes are restricted, as follows:

- No one can modify configuration attributes of the Generic server pool (all attributes are read-only)

- When DBCA or SRVCTL specifies a server name in the `HOSTING_MEMBERS` resource attribute, Oracle Clusterware only allows it if the server is:

  - Online and exists in the Generic server pool

  - Online and exists in the Free server pool, in which case Oracle Clusterware moves the server into the Generic server pool

  - Online and exists in any other server pool *and* the user is either a cluster administrator or is allowed to use the server pool's servers, in which case, the server is moved into the Generic server pool

  - Offline and the user is a cluster administrator

6.2 Oracle RAC Database and Server Pools

Oracle RAC databases support two different management styles and deployment models:

- **Policy-managed**: Deployment is based on server pools, where database services run within a server pool as *singleton* or *uniform* across all of the servers in the server pool. Databases are deployed in one or more server pools and the size of the server pools determine the number of database instances in the deployment. Policy management allows clusters and databases to expand or shrink as requirements change.

  A policy-managed database is defined by **cardinality**, which is the number of database instances you want running during normal operations. A policy-managed database runs in one or more database server pools that the cluster administrator creates in the cluster, and it can run on different servers at different times. A database instance starts on all servers that are in the server pools defined for the database.

  Clients can connect to a policy-managed database using the same SCAN-based connect string no matter which servers they happen to be running on at the time.

- **Administrator-managed**: Deployment is based on the Oracle RAC deployment types that existed before Oracle Database 11g release 2 (11.2) and requires that you statically configure each database instance to run on a specific node in the cluster, and that you configure database services to run on specific instances belonging to a certain database using the preferred and available designation.

  When you review the database resource for an administrator-managed database, you see a server pool defined with the same name as the Oracle database. This server pool is part of a special Oracle-defined server pool called Generic. Oracle RAC manages the Generic server pool to support administrator-managed databases. When you add or remove an administrator-managed database using either SRVCTL or DBCA, Oracle RAC creates or removes the server pools that are members of Generic.

  > **See Also:**
  >
  > - *Oracle Clusterware Administration and Deployment Guide* for detailed information about server pools
  > - *Oracle Clusterware Administration and Deployment Guide* for information about policy sets

6.3 About Creating Server Pools for Oracle RAC Databases

You can create a server pool with DBCA while creating an Oracle RAC database, but Oracle recommends that you create server pools before you deploy database software and databases. Oracle also recommends that you:

- Enable role separation before you create the first server pool in the cluster.

- Create and manage server pools using configuration policies and a respective policy set.

You can implement role-separated management in one of two ways:

- **Vertical implementation** (between layers) describes a role separation approach based on different operating system users and groups used for various layers in the technology stack. Permissions on server pools and resources are granted to different users (and groups) for each layer in the stack using access control lists. Oracle Automatic Storage Management (ASM) offers setting up role separation as part of the Oracle Grid Infrastructure installation based on a granular assignment of operating system groups for specific roles.

- **Horizontal implementation** (within one layer) describes a role separation approach that restricts resource access within one layer using access permissions for resources that are granted using access control lists assigned to server pools and policy-managed databases or applications.

  For example, consider an operating system user called `grid`, with primary operating system group `oinstall`, that installs Oracle Grid Infrastructure and creates two database server pools. The operating system users `ouser1` and `ouser2` must be able to operate within a server pool, but should not be able to modify those server pools and withdraw hardware resources from other server pools either accidentally or intentionally.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for information about creating policy sets

- *Oracle Clusterware Administration and Deployment Guide* for information about configuring role-separated management

6.4 Oracle RAC One Node and Server Pools

Note the following about Oracle RAC One Node and server pools:

- Oracle RAC One Node runs only in one server pool. This server pool is treated the same as any other server pool.

- Online relocation of an Oracle RAC One Node database instance permits planned migrations of an Oracle RAC One Node database from one node to another node. Relocations must always be within a server pool.

7

Understanding the Oracle RAC Installed Configuration

This chapter describes the Oracle Real Application Clusters (Oracle RAC) installed configuration. This chapter contains the following topics:

- Understanding the Configured Environment in Oracle RAC
- Understanding Operating System Privileges Groups
- Understanding Time Zone Settings on Cluster Nodes
- Understanding the Server Parameter File for Oracle RAC
- ORATAB Configuration for Oracle RAC
- Database Components Created Using Database Configuration Assistant
- Managing Undo Tablespaces in Oracle RAC
- Initialization Parameter Files
- Oracle Net Services Configuration for Oracle RAC Databases
- Performance Features of Oracle Net Services and Oracle RAC
- Oracle Net Services Configuration Files and Parameters

7.1 Understanding the Configured Environment in Oracle RAC

Oracle Net Configuration Assistant (NETCA) and Database Configuration Assistant (DBCA) configure your environment to meet the requirements for database creation and Oracle Enterprise Manager discovery of Oracle RAC databases.

> **Note:** Configuration files are created on each node in your cluster database.

Avoid changing host names after you complete your Oracle RAC installation, including adding or deleting domain qualifications. Node names are created from the host names during an Oracle Clusterware installation and are used extensively with database processes. Nodes with changed host names must be deleted from the cluster and added back with the new host names.

7.2 Understanding Operating System Privileges Groups

As an administrator, you often perform special operations such as shutting down or starting up a database, or configuring storage. Because only an administrator responsible for these administration decisions should perform these operations, system privileges for Oracle Database or Oracle Automatic Storage Management (Oracle ASM) administration require a secure authentication scheme.

Membership in special operating system groups enables administrators to authenticate to Oracle Database or Oracle ASM through the operating system rather than with a user name and password. This is known as **operating system authentication**. Each Oracle Database in a cluster can have its own operating system privileges groups, so that operating system authentication can be separated for each Oracle Database on a cluster. Because there can be only one Oracle Grid Infrastructure installation on a cluster, there can be only one set of operating system privileges groups for Oracle ASM.

During installation of Oracle Grid Infrastructure and Oracle Database, you provide the group names of operating system groups. These operating system groups are designated with the logical role of granting operating system group authentication for administration system privilege for Oracle Database and Oracle ASM.

In an Oracle RAC cluster, the group ID number (GID) for system privileges groups must be identical on each cluster member node. One operating system group can be designated the logical group whose members are granted all system privileges for Oracle Database and Oracle ASM, including the OINSTALL system privileges for installation owners. You can also delegate logical system privileges to two or more actual operating system groups. Oracle recommends that you designate separate operating system groups for each logical system privilege. This enables you to grant one or more subsets of administrator system privileges to database administrators. These database administrators can then perform standard database administration tasks without requiring the SYSDBA system privileges.

System privileges groups are listed in Table 7–1:

Table 7–1   Role-Allocated Oracle System Privileges Operating System Groups

Logical Operating System Group Name	Default Actual UNIX or Linux Group Name	System Privileges Authenticated By Group Membership
OINSTALL	oinstall	Install system privileges for installation owners, which includes privileges to write to the central oraInventory directory for each server, and other privileges granted to Oracle binary installation owner users.
OSDBA	dba	SYSDBA system privileges for an Oracle Database, which includes all system privileges for the database.
OSOPER	oper	SYSOPER startup and shutdown system privileges for an Oracle Database.
OSBACKUPDBA	backupdba	SYSBACKUP backup and recovery system privileges for an Oracle Database.
OSDGDBA	dgdba	SYSDG system privileges to administer and monitor Oracle Data Guard.
OSKMDBA	kmdba	SYSKM system privileges for encryption key management for applications such as Oracle Wallet Manager.

Table 7–1   (Cont.) Role-Allocated Oracle System Privileges Operating System Groups

Logical Operating System Group Name	Default Actual UNIX or Linux Group Name	System Privileges Authenticated By Group Membership
OSASM	asmadmin	SYSASM system privileges for Oracle ASM on a cluster, which includes all system privileges for Oracle ASM storage.
OSOPER for ASM	asmoper	SYSOPER startup and shutdown system privileges for Oracle ASM on the cluster.
OSDBA for ASM	asmdba	SYSDBA for ASM system privileges to obtain read and write access to files managed by Oracle ASM; all Oracle Database software owners must be a member of this group.

> **See Also:**
>
> - *Oracle Database Administrator's Guide* for more information about operating system groups and Oracle Database system privileges
> - *Oracle Automatic Storage Management Administrator's Guide* for more information about operating system groups and Oracle ASM system privileges

7.3 Understanding Time Zone Settings on Cluster Nodes

Oracle RAC requires that all cluster nodes have the same time zone setting. During an Oracle Grid Infrastructure for a cluster installation, the installation process determines the time zone setting of the Grid installation owner on the node where Oracle Universal Installer (OUI) runs. OUI uses that time zone value on all of the nodes as the default time zone setting for all processes that Oracle Clusterware manages. This default setting is used for databases, Oracle ASM, and any other managed processes. However, if you start an instance with SQL*Plus, you must ensure that the time zone value that Oracle RAC uses is the same as the Oracle Clusterware time zone. You can change the time zone that Oracle Clusterware uses for a database by running the command `srctl setenv database -env 'TZ=time zone'`

7.4 Understanding the Server Parameter File for Oracle RAC

When you create the database, Oracle Database creates an SPFILE in the file location that you specify. This location can be either an Oracle ASM disk group or a cluster file system.

All instances in the cluster database use the same SPFILE at startup. Because the SPFILE is a binary file, do not directly edit the SPFILE with an editor. Instead, change SPFILE parameter settings using Oracle Enterprise Manager or ALTER SYSTEM SQL statements.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* for information about creating and modifying SPFILE

7.5 ORATAB Configuration for Oracle RAC

Oracle creates an entry for each Oracle RAC database in the `oratab` configuration file. The `oratab` file is created by the `root.sh` script during installation, and it is updated by the Database Configuration Assistant when creating or deleting a database. The `oratab` file entry is also created automatically by the Database Agent when a database

is first started on a node where it has not run previously. Oracle ASM Agent creates Oracle ASM `oratab` entries.

Oracle Enterprise Manager uses the `oratab` file during service discovery to determine the name of the Oracle RAC database, and to determine if the database should be started automatically when the system is restarted.

The database entry has the following syntax:

```
$DB_UNIQUE_NAME:$ORACLE_HOME:N
```

A colon (:) is used as the field terminator. A new line terminates the entry. Lines beginning with a pound sign (#) are comments. Because all the instances of an Oracle RAC database have the same `DB_UNIQUE_NAME`, but each instance has its own `ORACLE_SID`, use the `$DB_UNIQUE_NAME` environment variable in the `oratab` file as the database entry.

The `$DB_UNIQUE_NAME` identifier for your Oracle RAC database must be unique across your enterprise. `$ORACLE_HOME` is the directory path to the database, and `N` indicates that the database should not be started at restart time. The following is an example entry for a database named `sales`:

```
sales:/u01/app/oracle/sales:N
```

7.6 Database Components Created Using Database Configuration Assistant

This section describes the database components that DBCA creates, which include:

- Tablespaces and Data files
- Control Files
- Online Redo Log Files

7.6.1 Tablespaces and Data files

For both single-instance and cluster database environments, Oracle Database is divided into smaller logical areas of space known as tablespaces. Each tablespace corresponds to one or more data files on the shared storage. Table 7–2 shows the tablespace names used by an Oracle RAC database and the types of data they contain.

Table 7–2   Tablespace Names Used with Oracle Real Application Clusters Databases

Tablespace Name	Contents
SYSAUX	An auxiliary system tablespace that contains the DRSYS (contains data for Oracle Text), CWMLITE (contains the OLAP schemas), XDB (for XML features), ODM (for Oracle Data Mining), INDEX and EXAMPLE schemas.
SYSTEM	Consists of the data dictionary, including definitions of tables, views, and stored procedures needed by the database. Oracle Database automatically maintains information in this tablespace.
TEMP	Contains temporary tables and indexes created during SQL statement processing. You may need to expand this tablespace if you run a SQL statement that involves significant sorting, such as ANALYZE COMPUTE STATISTICS on a very large table, or the constructs GROUP BY, ORDER BY, or DISTINCT.

Table 7–2   (Cont.)   Tablespace Names Used with Oracle Real Application Clusters Databases

Tablespace Name	Contents
UNDOTBS*n*	Contains undo tablespaces for each instance that DBCA creates for automatic undo management.
USERS	Consists of application data. As you create and enter data into tables, Oracle Database fills this space with your data.
EXAMPLE	Stores the sample schemas, if you opted to include them during database creation.

You cannot alter these tablespace names when using the preconfigured database configuration option from OUI. However, you can change the names of the tablespaces if you use the advanced database creation method.

As mentioned, each tablespace has one or more data files on shared file systems. The data file names created by the preconfigured database configuration options vary by storage type such as Oracle ASM, or a cluster file system.

7.6.2 Control Files

The database is configured with two control files that must be stored on shared storage. Every database must have one unique control file; any additional control files configured for the database are identical copies of the original control file.

If a control file becomes unusable, then the database instance fails when it attempts to access the damaged control file. By multiplexing (creating multiple copies of) a control file on different disks, the database can achieve redundancy and thereby avoid a single point of failure.

> **See Also:**
>
> - *Oracle Database Concepts*
> - *Oracle Database Administrator's Guide*

7.6.3 Online Redo Log Files

Each database instance must have at least two online redo log files. The online redo log files for a database instance are called the **redo thread**. Each Oracle RAC database instance has its own redo thread to avoid contention for a single set of online redo log files. In case of instance failure, the online redo log files must be accessible by the surviving instances. Therefore, the online redo log files for an Oracle RAC database must be placed on shared storage or Oracle ASM. If you use a file system for storage, then the file system must be a shared or cluster file system.

The file names of the redo log files that are created with the preconfigured database configuration option vary by storage type.

> **See Also:**
>
> - *Oracle Database Concepts* for more information about the online redo log files
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about storage for online redo log files

7.7 Managing Undo Tablespaces in Oracle RAC

Oracle Database stores rollback or undo information in undo tablespaces. To manage undo tablespaces, Oracle recommends that you use Automatic Undo Management. Automatic Undo Management is an automated management mode for undo tablespaces that is easier to administer than manual undo management.

When Oracle ASM and Oracle Managed Files are used along with Automatic Undo Management, an instance that is started for the first time, and thus does not have an undo tablespace, has its undo tablespace created for it by another instance automatically. The same is also true for redo logs.

> **See Also:**
>
> - *Oracle Database Administrator's Guide* for more information about automatic undo management
>
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about managing undo tablespaces

7.8 Initialization Parameter Files

Oracle recommends using the server parameter file (SPFILE) for storing Oracle Database initialization parameters. Oracle recommends that you store all SPFILEs on Oracle ASM, including the Oracle ASM SPFILE. SPFILEs must be located on shared storage; all instances in a cluster database can access this parameter file.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* for more information about the creation and use of parameter files

7.9 Oracle Net Services Configuration for Oracle RAC Databases

Users can access an Oracle RAC database using a client/server configuration or through one or more middle tiers, with or without connection pooling. When connecting to an Oracle Database, you can use a connect descriptor or a net service name. For Oracle RAC databases, you can also use the Single Client Access Name (SCAN) to connect to any available instance of the Oracle RAC database.

This section contains the following topics:

- Database Services for an Oracle RAC Database

- Naming Methods and Connect Descriptors

- Easy Connect Naming Method

- Understanding SCANs

- How to Connect to an Oracle RAC Database Using SCANs

- Listener Configuration for an Oracle RAC Database

- Service Registration for an Oracle RAC Database

- How Database Connections are Created When Using SCANs

> **See Also:** *Oracle Database Net Services Administrator's Guide* for more information about Oracle Net Services concepts

7.9.1 Database Services for an Oracle RAC Database

Each database is represented by one or more services. A service is identified by a service name, such as `sales.example.com`. A client uses a service name to identify the database it must access. During installation, Oracle RAC databases are configured with a default database service that has the same name as the database. This service can be used for performing database management tasks. Additional services should be created for client and application connections to the database.

A service name can be associated with multiple database instances, and an instance can be associated with multiple services. The listener acts as a mediator between the client and database instances and routes the connection request to the appropriate instance. Clients connecting to a service do not have to specify which instance they want to connect to.

7.9.2 Naming Methods and Connect Descriptors

Each net service name is associated with a connect descriptor. A **connect descriptor** provides the location of the database and the name of the database service. A connect descriptor is comprised of one or more protocol addresses of the listener and the connect information for the destination service.

The information needed to use a service name to create a database connection can be stored in a repository, which is represented by one or more naming methods. A **naming method** is a resolution method used by a client application to resolve a service name to a connect descriptor. Oracle Net Services offers several types of naming methods that support localized configuration on each client, or centralized configuration that can be accessed by all clients in the network.

7.9.3 Easy Connect Naming Method

The Easy Connect naming method eliminates the need to look up service names in the `tnsnames.ora` file or other repository for TCP/IP environments. With Easy Connect, clients use a connect string for a simple TCP/IP address, which consists of a host name, and an optional port and service name. If you use this method, then no naming or directory system is required. See Example 7–1, "Connecting to Oracle RAC Using the Easy Connect Naming Method" for an example.

Networking elements for the Oracle Database server and clients are preconfigured for most environments. The Easy Connect naming method is enabled by default and does not require a repository. If you use a naming method other than Easy Connect, then additional configuration of Oracle Net Services may be required.

7.9.4 Understanding SCANs

This section contains the following topics:

- About the SCAN
- SCAN VIP Addresses
- SCAN Listeners

7.9.4.1 About the SCAN

The SCAN is a domain name registered to at least one and up to three IP addresses, either in domain name service (DNS) or in Grid Naming Service (GNS). During the installation of Oracle Grid Infrastructure, several Oracle Clusterware resources are created for the SCAN:

- A SCAN virtual IP (VIP) is created for each IP address that SCAN resolves to

- A SCAN listener is created for each SCAN VIP

- A dependency on the SCAN VIP is configured for the SCAN listener

SCANs are defined using one of two options:

- The SCAN is defined in DNS

  If you configure a SCAN manually, and use DNS for name resolution, then your network administrator should create a single name for the SCAN that resolves to three IP addresses on the same network as the public network for the cluster. The SCAN name must be resolvable without the domain suffix (for example, the address `sales1-scan.example.com` must be resolvable using `sales1-scan`). The SCAN must not be assigned to a network interface, because Oracle Clusterware resolves the SCAN.

  The default SCAN is `cluster_name-scan.domain_name`. For example, in a cluster that does not use GNS, if your cluster name is `sales1`, and your domain is `example.com`, then the default SCAN address is `sales1-scan.example.com:1521`.

- The SCAN is defined in GNS

  When using GNS and DHCP, Oracle Clusterware configures the VIP addresses for the SCAN name that is provided during cluster configuration. The node VIP and the three SCAN VIPs are obtained from the DHCP server when using GNS. If a new server joins the cluster, then Oracle Clusterware dynamically obtains the required VIP address from the DHCP server, updates the cluster resource, and makes the server accessible through GNS.

Oracle recommends that you configure clients connecting to the cluster to use the SCAN name, rather than node VIPs used in releases before Oracle Grid Infrastructure 11*g* Release 2 (11.2). Clients connecting to Oracle RAC databases using SCANs do not have to be configured with addresses of each node that hosts a particular database or database instance. For example, if you configure policy-managed server pools for a cluster, then connecting to the database using a SCAN enables connections to server pools in that database, regardless of which nodes are allocated to the server pool. You can add or remove nodes from the database without having to reconfigure clients connecting to the database.

> **See Also:** *Oracle Grid Infrastructure Installation Guide* for your platform for more information about SCAN configuration and requirements

7.9.4.2 SCAN VIP Addresses

SCAN virtual IP addresses (VIPs) function like node VIPs. However, unlike node VIPs, SCAN VIPs can run on any node in the cluster. Clients (users or applications) that connect using a SCAN instead of a node VIP name or address do not have to update the list of node names or addresses in their local `tnsnames.ora` file when nodes are added to or removed from the cluster, or when a database instance runs on a different node.

> **Note:** Configuring three SCAN VIPs in DNS does not by itself ensure failover of connections. Instead, the Oracle Client uses the returned SCAN VIPs to failover the connection request to a different SCAN listener. If the connection attempt to a SCAN VIP fails, then the client uses the next returned SCAN VIP address to connect. For this reason, Oracle recommends that you use Oracle Client 11*g* Release 2 or later clients for connections that use the SCAN.

If you use GNS for name resolution, then you only provide the SCAN name during installation (for example, `sales1-scan`). GNS obtains DHCP address leases for three IP addresses and resolves these addresses to the SCAN. The GNS daemon listens for registrations. When a SCAN VIP starts on a node, it registers its addresses with GNS.

Service requests to the cluster domain that GNS manages are routed to the GNS VIP address, which routes these requests to the GNS daemon for the cluster. When GNS receives a request from a DNS for the SCAN, it returns the registered addresses of the SCAN listeners to the DNS. The DNS then returns the three SCAN VIP addresses to the client.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for more information about SCAN names, listeners, and client service requests

7.9.4.3 SCAN Listeners

During Oracle Grid Infrastructure installation, SCAN listeners are created for as many IP addresses as there are SCAN VIP addresses assigned to resolve to the SCAN. Oracle recommends that the SCAN resolves to three VIP addresses, to provide high availability and scalability. If the SCAN resolves to three addresses, then three SCAN VIPs and three SCAN listeners are created.

Each SCAN listener depends on its corresponding SCAN VIP. The SCAN listeners cannot start until the SCAN VIP is available on a node.

The addresses for the SCAN listeners resolve either through an external domain name service (DNS), or through the Grid Naming Service (GNS) within the cluster. SCAN listeners and SCAN VIPs can run on any node in the cluster. If a node where a SCAN VIP is running fails, then the SCAN VIP and its associated listener fails over to another node in the cluster. If the number of available nodes within the cluster falls to less than three, then one server hosts two SCAN VIPs and SCAN listeners.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for more information about SCAN listeners

7.9.5 How to Connect to an Oracle RAC Database Using SCANs

Oracle recommends that you configure Oracle RAC database clients to use the SCAN and the Easy Connect naming method to connect to the database instead of configuring the `tnsnames.ora` file.

Clients configured to connect to the cluster using node VIP addresses for Oracle RAC releases earlier than Oracle Database 11*g* Release 2 can continue to use their existing connection addresses; using the SCAN is not required. When an earlier release of Oracle Database is upgraded, the database is not only registered with the local listeners, but is also registered with the SCAN listeners, allowing clients to start using the SCAN to connect to that database.

Example 7–1   Connecting to Oracle RAC Using the Easy Connect Naming Method

If the Oracle RAC database runs on a cluster for which the SCAN is `sales1-scan.mycluster.example.com`, then you can submit a connection request for the database service `oltp.example.com` by using a connect descriptor similar to the following:

```
sqlplus system/manager@sales1-scan.mycluster.example.com:1521/oltp
```

If the SCAN is resolved by DNS, then DNS returns all three SCAN VIP addresses in round-robin order to the client. If the SCAN is resolved by GNS, then DNS zone delegation sends the lookup request to GNS, which then returns all three SCAN VIP addresses in round-robin order to the client.

The client then uses one of the returned SCAN VIP addresses to contact a SCAN listener. When a SCAN listener receives a connection request from a client, the SCAN listener identifies the least loaded instance in the cluster that provides the requested service. It then redirects the connection request to the local listener on the node where the least loaded instance is running, and the client is given the local listener address. The local listener then creates the connection to the database instance.

7.9.6 Listener Configuration for an Oracle RAC Database

An Oracle database receives connection requests through the local listener. The local listener brokers a client request, handing off the request to the server. The listener is configured with a protocol address, and clients configured with the same protocol address can send connection requests to the listener. When a connection is established, the client and Oracle database communicate directly with one another.

The local listener, or default listener, is located in the Grid home when you have Oracle Grid Infrastructure installed. Local listeners are configured to respond to database connection requests, and to non-database connection requests, such as external procedures or Oracle XML Database (XDB) requests. When the database starts, the Database Agent process (`oraagent`, previously known as `racgimon`) sets the `LOCAL_LISTENER` parameter to a connect descriptor that does not require an Oracle Net service name. The value for `LOCAL_LISTENER` is computed to be the endpoints of the Grid home listeners.

You can configure multiple Oracle Database listeners, each with a unique name, in one `listener.ora` file. Multiple listener configurations for database listeners are possible because each of the top-level configuration parameters has a suffix of the listener name or is the listener name itself. To configure a database to register with multiple local listeners, you must manually modify the `LOCAL_LISTENER` parameter.

> **Note:**   Oracle recommends running only one listener for each node in most customer environments.

For an Oracle RAC database, the database parameter `REMOTE_LISTENER` identifies the SCAN listeners. The database registers with the local and SCAN listeners by using the connect description information contained in these parameters. Oracle Database 11*g* Release 2 and later instances only register with SCAN listeners as remote listeners. Upgraded databases register with SCAN listeners as remote listeners, and also continue to register with all node listeners.

The `REMOTE_LISTENER` parameter for an Oracle RAC database is always set to the SCAN address. For example, if the SCAN for the cluster is `myscan`, and the GNS

subdomain for the cluster is `mycluster.example.com`, then the `REMOTE_LISTENER` parameter has the following value:

`myscan.mycluster.example.com:1521`

> **Note:** Do not set the `REMOTE_LISTENER` parameter for an Oracle RAC database to an Oracle Net alias that has a single address that uses the SCAN for the host name (`HOST=scan`).

7.9.7 Service Registration for an Oracle RAC Database

An Oracle Database 12*c* Release 1 (12.1) database service automatically registers with the listeners specified in the database initialization parameters `LOCAL_LISTENER` and `REMOTE_LISTENER`. During registration, the listener registration (LREG) process sends information such as the service name, instance names, and workload information to the listeners. This feature is called **service registration**.

When a listener starts after the Oracle instance starts, and the listener is available for service registration, registration does not occur until the next time the Oracle Database LREG process starts its discovery routine. By default, the LREG discovery routine is started every 60 seconds. To override the 60-second delay, use the SQL statement `ALTER SYSTEM REGISTER`. This statement forces LREG to register the service immediately.

> **Note:** Oracle recommends that you create a script to run the `ALTER SYSTEM REGISTER` statement immediately after starting the listener. If you run this statement when the instance is registered and all services are currently registered, or while the listener is down, then the statement has no effect.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for more information about service registration

7.9.8 How Database Connections are Created When Using SCANs

Based on the environment, the following actions occur when you use a SCAN to connect to an Oracle RAC database using a service name. The numbered actions correspond to the arrows shown in Figure 7–1.

1. The LREG process of each instance registers the database services with the default listener on the local node and with each SCAN listener, which is specified by the `REMOTE_LISTENER` database parameter. The listeners are dynamically updated on the amount of work being handled by the instances and dispatchers.

2. The client issues a database connection request using a connect descriptor of the form:

    `orausr/@scan_name:1521/webapp`

> **Note:** If you use the Easy Connect naming method, then ensure that the `sqlnet.ora` file on the client contains `EZCONNECT` in the list of naming methods specified by the `NAMES.DIRECTORY_PATH` parameter.

3. The client uses DNS to resolve *scan_name*. After DNS returns the three addresses assigned to the SCAN, the client sends a connect request to the first IP address. If the connect request fails, then the client attempts to connect using the next IP address.

4. When the connect request is successful, the client connects to a SCAN listener for the cluster that hosts the `sales` database and has an instance offering the `webapp` service, which in this example is `sales1` and `sales2`. The SCAN listener compares the workload of the instances `sales1` and `sales2` and the workload of the nodes on which they run. If the SCAN listener determines that `node2` is less loaded than `node1`, then the SCAN listener selects `node2` and sends the address for the local listener on that node back to the client.

5. The client connects to the local listener on `node2`. The local listener starts a dedicated server process for the connection to the database.

6. The client connects directly to the dedicated server process on `node2` and accesses the `sales2` database instance.

Figure 7–1 Load Balancing Actions for Oracle RAC Connections That Use SCAN
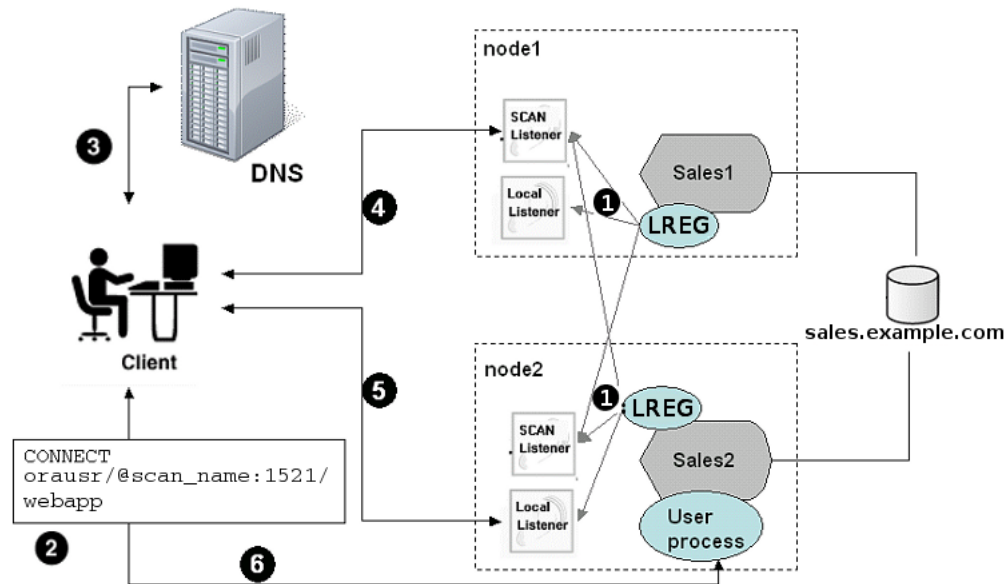
Figure 7-1 is an illustration of how load balancing works on an Oracle RAC system when SCAN is used. It contains 5 callout steps, and one visual presentation of an action. Step 1 is in the center right of the frame, corresponding with step 1 of the preceding description. Step 2 is in the lower left side of the frame, corresponding with step 2 of the preceding description. Step 3 is in the upper right side of the image, corresponding with step 3 of the description. Step 4 is in the upper center of the image, corresponding with step 4 of the description. Step 5 is in the lower center of the image, corresponding with step 5 of the description. There is no callout for step 6 of the description, The client is represented by the image of a person connecting to the database through a terminal, on the left center of the image, and the database instance on node 2 is represented on the lower right side of the image. There is a line between the client and the database instance on node 2, representing the connection between the client and the database.

****

7.10 Performance Features of Oracle Net Services and Oracle RAC

Oracle RAC databases provide the important benefits of connection load balancing and failover.

- Load Balancing of Connections to Oracle RAC Databases
- Connection Failover for Oracle RAC Databases
- Shared Server Configuration for an Oracle RAC Database

7.10.1 Load Balancing of Connections to Oracle RAC Databases

Services coordinate their sessions by registering their **workload**, or the amount of work they are currently handling, with the local listener and the SCAN listeners. Clients are redirected by the SCAN listener to a local listener on the least-loaded node that is running the instance for a particular service. This feature is called **load balancing**. The local listener either directs the client to a dispatcher process (if you configured the database to use shared servers), or directs the client to a dedicated server process.

There are two types of load balancing that you can implement for an Oracle RAC database: client-side and server-side load balancing. Client-side load balancing balances the connection requests across the listeners. With server-side load balancing, the SCAN listener directs a connection request to the best instance currently providing the service by using the load balancing advisory.

> **See Also:**
>
> - *Oracle Grid Infrastructure Installation Guide* for more information about SCAN and its configuration
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about failover, load balancing, and the load balancing advisory

7.10.2 Connection Failover for Oracle RAC Databases

When a client issues a connection request using SCAN, the three SCAN addresses are returned to the client. If the first address fails, then the connection request to the SCAN fails over to the next address. Using multiple addresses allows a client to connect to an instance of the database even if the initial instance has failed.

Oracle RAC provides failover with the node VIP addresses by configuring multiple listeners on multiple nodes to manage client connection requests for the same database service. If a node fails, then the service connecting to the VIP is relocated transparently to a surviving node, enabling fast notification of the failure to the clients connecting through the VIP. If the application and client are configured with transparent application failover options, then the client is reconnected to the surviving node.

7.10.3 Shared Server Configuration for an Oracle RAC Database

Standalone Oracle databases perform load balancing by distributing connections among the shared server dispatcher processes. By default, DBCA configures your Oracle RAC database with dedicated servers, not shared servers. However, if you select the shared server option when using DBCA, then DBCA configures shared servers. Oracle RAC uses both dedicated and shared server processing when shared servers are configured.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for more information about shared and dedicated server configurations

7.11 Oracle Net Services Configuration Files and Parameters

Networking elements for the Oracle Database server and clients are preconfigured for most environments. The Easy Connect naming method is enabled by default and does not require a repository. If you use a naming method other than Easy Connect, then additional configuration of Oracle Net Services may be required.

The following sections describe the Oracle Net Services configuration files and parameters for an Oracle RAC database:

- Database Initialization Parameters for Database Service Registration
- Net Service Names and the tnsnames.ora File
- Net Service Names Created by DBCA
- Listener Configuration and the listener.ora File
- Net Services Profile File (sqlnet.ora)

7.11.1 Database Initialization Parameters for Database Service Registration

An Oracle Database 12*c* Release 1 (12.1) database service automatically registers with the listeners specified in the `LOCAL_LISTENER` and `REMOTE_LISTENER` parameters. During registration, the listener registration (LREG) process sends information such as the service name, instance names, and workload information to the listeners.

When a listener starts after the Oracle instance starts, and the listener is available for service registration, registration does not occur until the next time the Oracle Database LREG process starts its discovery routine. By default, the LREG discovery routine is started every 60 seconds. To override the 60-second delay, use the SQL statement `ALTER SYSTEM REGISTER`. This statement forces LREG to register the service immediately.

> **Note:** Oracle recommends that you create a script to run the `ALTER SYSTEM REGISTER` statement immediately after starting the listener. If you run this statement when the instance is registered and all services are currently registered, or while the listener is down, then the statement has no effect.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for more information about service registration

7.11.2 Net Service Names and the tnsnames.ora File

The installation process creates a `tnsnames.ora` file on each node. This file acts as a repository of net service names. Each net service name is associated with a connect identifier. A **connect identifier** is an identifier that maps a user-defined name to a connect descriptor. A connect descriptor contains the following information:

- The network route to the service, including the location of the listener through a protocol address
- The `SERVICE_NAME` parameter, with the value set to the name of a database service

> **Note:** The SERVICE_NAME parameter that you use in the
> tnsnames.ora file is singular, because you can specify only one
> service name. The SERVICE_NAME parameter is not the same as the
> service_names database initialization parameter. The service_
> names database parameter defaults to the global database name, a
> name comprising the db_name and db_domain parameters in the
> initialization parameter file. When you add service names using
> SRVCTL or Oracle Enterprise Manager Cloud Control, it lists
> additional cluster-managed services for the database.

The tnsnames.ora file is located in both the *Grid_home*/network/admin and *Oracle_
home*/network/admin directories. By default, the tnsnames.ora file is read from the
Grid home when Oracle Grid Infrastructure is installed.

With Oracle Clusterware 11*g* Release 2 and later, the listener association no longer
requires tnsnames.ora file entries. The listener associations are configured as follows:

- DBCA no longer sets the LOCAL_LISTENER parameter. The Oracle Clusterware
  agent that starts the database sets the LOCAL_LISTENER parameter dynamically, and
  it sets it to the actual value, not an alias. So listener_*alias* entries are no longer
  needed in the tnsnames.ora file.

- The REMOTE_LISTENER parameter is configured by DBCA to reference the SCAN
  and SCAN port, without any need for a tnsnames.ora entry. Oracle Clusterware
  uses the Easy Connect naming method with *scanname*:*scanport*, so no listener
  associations for the REMOTE_LISTENER parameter are needed in the tnsnames.ora
  file.

For example, after you create the database, to add a second listener, listening on port
2012, use a command similar to the following command to have the database register
with both listeners on startup:

```
SQL> alter system set local_listener='(DESCRIPTION=
(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.0.61)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.0.61)(PORT=2012))))'
scope=BOTH SID='OCRL1';
```

> **See Also:**
>
> - *Oracle Database Administrator's Guide*
> - *Oracle Database Net Services Administrator's Guide* for more
>   information about the tnsnames.ora file

7.11.3 Net Service Names Created by DBCA

DBCA creates net service names for connections as described in the following sections:

- Net Service Names for Database Connections
- Net Service Names for Instance Connections

7.11.3.1 Net Service Names for Database Connections

Clients that connect to any instance of Oracle RAC use the SCAN in the connect
descriptor. You can also use net service names to connect to Oracle RAC. The default
database service created by DBCA enables Oracle Enterprise Manager to discover an
Oracle RAC database, and should not be used for client connections.

If you use DBCA to create an Oracle RAC database that is a multitenant container database (CDB), then DBCA creates a database service that has the same name as the database. Clients that use this database service can connect to any database instance for the Oracle RAC CDB. However, if you use DBCA to add a pluggable database (PDB) to an existing CDB, then DBCA does not create a database service for the new PDB.

Example 7–2   Example Net Service Name Entry for a Database Connection

This example shows a connect descriptor that is used in a `tnsnames.ora` file. The connect identifier in this case is the same as the cluster domain, `mycluster.example.com`. Instead of specifying the address for an individual server, virtual Internet Protocol (VIP) address, or a cluster node name, the connect descriptor uses the SCAN, which is `myscan.mycluster.example.com`.

```
mycluster.example.com =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = host=myscan.mycluster.example.com)
      (PORT = 1522))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = myApp)
    )
  )
```

Oracle Clusterware resolves connection requests that use the net service name `mycluster.example.com` to any of the database instances of the `mycluster` database that run the `myApp` database service. The specific cluster node on which the instance is running is invisible to the client.

The net service name does not require a fully qualified domain name for the server on which a database, database instance, or listener runs. SCANs are resolved by the DNS or GNS, which returns three addresses to the client. The client then submits connection requests to each address in succession until a connection is made.

7.11.3.2 Net Service Names for Instance Connections

Clients that connect to a particular instance of the database use the net service name for the instance.

Example 7–3   Example Net Service Name Entry for an Instance Connection

In this example, the connect identifier is the same as the instance name, `mycluster1.example.com`. The connect descriptor uses the SCAN to locate the instance within the cluster. Clients connecting to the net service name `mycluster1.example.com` are connected to the `mycluster1` database instance of the `mycluster` database. Oracle Clusterware resolves that connection to the cluster node on which the instance is running. The specific cluster node on which the instance is running is invisible to the client.

```
mycluster1.example.com=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=TCP)(HOST=myscan.mycluster.example.com)(PORT=1521))
    (CONNECT_DATA=
      (SERVICE_NAME=mycluster.example.com)
      (INSTANCE_NAME=mycluster1)
    )
  )
```

7.11.4 Listener Configuration and the listener.ora File

In Oracle RAC environments, Oracle recommends that you let the Oracle Agent manage Oracle listeners for Oracle Databases. The following sections describe Oracle Net listener configuration:

- Local Listener for an Oracle RAC Database
- Remote Listeners for an Oracle RAC Database
- Managing Multiple Listeners for an Oracle RAC Database
- How Oracle Database Uses the Listener File (listener.ora)

> **Note:** If you enable GNS, then you do not have to manually configure the listener.

7.11.4.1 Local Listener for an Oracle RAC Database

The local listener, or default listener, is located in the Grid home when you have Oracle Grid Infrastructure installed. The `listener.ora` file is located in the *Grid_home*`/network/admin` directory. If needed, you can edit the `listener.ora` file for the Grid home listeners to define listener parameters for node and SCAN listeners. Do not modify the endpoints because these are automatically managed by the listener agent.

During Oracle Database creation, the `LOCAL_LISTENER` parameter is automatically configured to point to the local listener for the database. You can set a value manually for `LOCAL_LISTENER`. If you modify the value of the `LOCAL_LISTENER` parameter, then the Database Agent process does not automatically update this value. Oracle recommends that you leave the parameter unset so that the Database Agent process can maintain it automatically. If you do not set `LOCAL_LISTENER`, then the Database Agent process automatically updates the database associated with the local listener in the Grid home, even when the ports or IP address of that listener are changed.

> **See Also:**
>
> - "Net Service Names and the tnsnames.ora File" on page 7-14 for more information about listener associations defined in the `tnsnames.ora` file
> - *Oracle Database Net Services Reference* for more information about the `listener.ora` file
> - *Oracle Database Net Services Administrator's Guide* for information about understanding and configuring listeners

7.11.4.2 Remote Listeners for an Oracle RAC Database

A **remote listener** is a listener residing on one computer that redirects connections to a database instance on another computer. For example, SCAN listeners are remote listeners. In Oracle RAC environments, Oracle recommends that you let the Oracle Agent manage the Oracle listeners for the databases.

See Also:

- "Net Service Names and the tnsnames.ora File" on page 7-14 for more information about listener associations defined in the `tnsnames.ora` file

- *Oracle Database Net Services Reference* for more information about the `listener.ora` file

- *Oracle Database Net Services Reference* for an overview of listeners

7.11.4.3 Managing Multiple Listeners for an Oracle RAC Database

To administer Oracle Database 12*c* Release 1 (12.1) local and SCAN listeners using the `lsnrctl` command, set your `ORACLE_HOME` environment variable to the path for the Grid home. Do not attempt to use the `lsnrctl` commands from Oracle home locations for earlier releases, because they cannot be used with Oracle Database 12*c* Release 1 (12.1).

For listeners not managed by Oracle Clusterware, you can use a non-default location for the `listener.ora` file by setting the `TNS_ADMIN` environment variable or registry value to point to the directory that contains the Oracle Net Services configuration files. To use a non-default location for a listener managed by Oracle Clusterware, you must use SRVCTL and the `setenv` command to modify the value of `TNS_ADMIN` for each listener.

7.11.4.4 How Oracle Database Uses the Listener File (listener.ora)

The `listener.ora` file is the configuration file for a listener. It can include the protocol addresses it is accepting connection requests on, a list of the database and other services it is listening for, and control parameters used by the listener. You can modify the configuration of the listeners used by Oracle Clusterware and Oracle RAC with Server Control Utility (SRVCTL) commands, or by using NETCA. Manual editing of the `listener.ora` file is not required.

Each listener is configured with one or more protocol addresses that specify its listening endpoints. The listener agent dynamically updates endpoints with the listener. Starting with Oracle Database 11*g* Release 2, the `listener.ora` file now only contains an IPC key and the following information:

```
(ADDRESS = (PROTOCOL=TCP)(HOST=)(PORT=1521))
```

In the previous example, the protocol `ADDRESS` refers implicitly to the `HOST` endpoint of the local node. The `listener.ora` file is the same on every node for an Oracle RAC database. Listening endpoints, such as the port numbers, are dynamically registered with the listener.

Before you install Oracle RAC, during the Oracle Grid Infrastructure installation, NETCA creates and starts a default listener in the Grid home called `LISTENER`. The listener is configured with default protocol listening addresses. The listener is configured to respond to connection requests that are directed to one protocol address specified during installation.

During the Oracle RAC installation, the Oracle RAC database uses the listener in the Grid home, and configures service information about the Oracle RAC database. The database services automatically register their information with the listener, such as the service name, instance names, and load information. Dynamic service registration eliminates the need for static configuration of database services. However, static service configuration is required if you plan to use Oracle Enterprise Manager.

Example 7–4   Example listener.ora File for an Oracle RAC Node

The following is an example listener.ora file as it would appear after installation, with an entry for a node named node1 and a SCAN listener.

```
LISTENER_SCAN1=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER_
SCAN1))))                # line added by Agent
LISTENER_NODE1=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)
(KEY=LISTENER))))
          # line added by Agent
# listener.ora.mycluster Network Configuration File:
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora.mycluster
# Generated by Oracle configuration tools.

LISTENER_NODE1 =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_NODE1=ON# line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_SCAN2=ON# line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_SCAN1=ON# line added by Agent
```

7.11.5 Net Services Profile File (sqlnet.ora)

Oracle Universal Installer starts Oracle Net Configuration Assistant (NETCA) after the database software is installed. NETCA creates the Oracle Net Services profile, or the sqlnet.ora file. In an Oracle Grid Infrastructure installation, the sqlnet.ora file is located in the following directory by default:

Grid_home/network/admin

For the local listener for the Oracle RAC database instance, the default location of the sqlnet.ora file is $ORACLE_HOME/network/admin directory. In this directory there is a default sqlnet.ora file. Also, you can find a sample sqlnet.ora file in the subdirectory sample.

During installation of the Oracle RAC software, NETCA creates the following entry in the sqlnet.ora file, where $ORACLE_BASE is the path to the Oracle base directory for the Oracle RAC installation:

```
NAMES.DIRECTORY_PATH=(TNSNAMES, EZCONNECT)
ADR_BASE =$ORACLE_BASE
```

The parameter NAMES.DIRECTORY_PATH specifies the priority order of the naming methods to use to resolve connect identifiers to connect descriptors.

The ADR_BASE parameter specifies the base directory into which tracing and logging incidents are stored when Automatic Diagnostic Repository (ADR) is enabled for the database.

See Also:

- *Oracle Database Net Services Administrator's Guide* for more information about the `sqlnet.ora` file

- *Oracle Database Concepts* for more information about authentication

- *Oracle Database Administrator's Guide* for more information about ADR

8

Removing Oracle Real Application Clusters Software

This chapter describes how to completely remove all Oracle databases, instances, and software from an Oracle Real Application Clusters (Oracle RAC) Oracle home directory.

This chapter contains the following topics:

- Overview of Deinstallation Procedures
- Identifying All Instances On a Cluster
- Deinstalling Oracle RAC Software

> **See Also:** The product-specific documentation for removing a single product to ensure that you are aware of requirements and restrictions that apply for that product

8.1 Overview of Deinstallation Procedures

To completely remove all Oracle databases, instances, and software from an Oracle home directory:

- Identify all instances associated with the Oracle home
- Shut down processes
- Remove listeners installed in the Oracle Database home
- Remove database instances
- Remove Oracle Automatic Storage Management Release 1 (11.1) or earlier
- Remove Oracle Clusterware and Oracle Automatic Storage Management (Oracle Grid Infrastructure)

> **Note:** For information on removing Oracle Database Vault, see *Oracle Database Vault Administrator's Guide*.
>
> For information on removing Oracle Configuration Manager, see *Oracle Configuration Manager Installation and Administration Guide*.
>
> With Oracle Grid Infrastructure Release 11.2 and later, Oracle Automatic Storage Management and Oracle Clusterware comprise the Oracle Grid Infrastructure installation. These components are installed and removed together.

> **Caution:** If any cluster member node has more than one database with the same global database name (GDN) on a server, then you cannot use the deinstall tool to remove one database only. For example, if you have a standalone database on one of your cluster nodes with the GDN `mydb.example.com`, and your Oracle RAC database GDN is also `mydb.example.com`, then both databases on that node are removed.

8.2 Identifying All Instances On a Cluster

To identify all instances associated with the Oracle home to remove, enter the following command, where *dbname* is the name of the database:

```
$ srvctl status database -d dbname
```

Alternately, you can check for registered instances by viewing the `oratab` file:

AIX, HP-UX, or Linux:

```
$ more /etc/oratab
```

Oracle Solaris:

```
$ more /var/opt/oracle/oratab
```

The output of this command contains entries similar to the following:

```
+ASM1:/u01/app/12.1.0/grid:N
CUST:/u01/app/oracle/product/12.1.0/dbhome_1:N
```

These entries show that the `+ASM` Oracle Automatic Storage Management instance in the Oracle Grid Infrastructure for a cluster home (`/u01/app/12.1.0/grid`) and the `CUST` Oracle database instance are associated with the Oracle home directory `/u01/app/oracle/product/12.1.0/dbhome_1`.

8.3 Deinstalling Oracle RAC Software

The Deinstallation Tool removes Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM) from your server, and Oracle Real Application Clusters (Oracle RAC) or standalone Oracle Database installations. The following sections describe the script, and provide information about additional options to use the tool:

- About the Deinstallation Tool
- Example of Running the Deinstall Command for Oracle RAC
- Deinstallation Parameter File Example for Oracle RAC

8.3.1 About the Deinstallation Tool

Starting with Oracle Database 12*c*, the deinstallation tool is integrated with the database installation media. You can run the deinstallation tool using the `runInstaller` command with the `-deinstall` and `-home` options from the base directory of the database, client or grid infrastructure installation media.

The deinstallation tool is also available as a separate command (`deinstall`) in Oracle home directories after installation. It is located in the `$ORACLE_HOME/deinstall` directory.

The deinstallation tool uses the information you provide, plus information gathered from the software home to create a parameter file. You can alternatively supply a parameter file generated previously by the deinstall command using the -checkonly option, or by editing the response file template.

The deinstallation tool stops Oracle software, and removes Oracle software and configuration files on the operating system for a specific Oracle home. If you run the deinstallation tool to remove Oracle Grid Infrastructure, then the deinstaller prompts you to run the rootcrs.sh script, as the root user, to deconfigure Oracle Grid Infrastructure or roothas.sh script to deconfigure Oracle Grid Infrastructure for standalone server.

> **Caution:** When you run the deinstallation tool, if the central inventory (oraInventory) contains no other registered homes besides the home that you are deconfiguring and removing, then the deinstall command removes the following files and directory contents in the Oracle base directory of the Oracle Database installation owner:
>
> - admin
> - cfgtoollogs
> - checkpoints
> - diag
> - oradata
> - fast_recovery_area
>
> Oracle strongly recommends that you configure your installations using an Optimal Flexible Architecture (OFA) configuration, and that you reserve Oracle base and Oracle home paths for exclusive use of Oracle software. If you have any user data in these locations in the Oracle base that is owned by the user account that owns the Oracle software, then the deinstallation tool deletes this data.

The default method for running the deinstallation tool is from the deinstall directory in the Oracle home as the installation owner:

```
$ $ORACLE_HOME/deinstall/deinstall
```

The deinstall command uses the following syntax, where variable content is indicated in italics:

```
deinstall [-silent] [-checkonly] [-local] [-paramfile complete path of input
response file]
[-params name1=value name2=value . . .] [-o complete path of directory for saving
files] [-help]
```

To run the deinstallation tool from the database installation media, use the runInstaller command with the -deinstall option, followed by the -home option to specify the path of the Oracle home to remove using the following syntax, where variable content is indicated in italics:

```
runInstaller -deinstall -home complete path of Oracle home [-silent] [-checkonly]
[-local] [-paramfile complete path of input parameter property file] [-params
name1=value
name2=value . . .] [-o complete path of directory for saving files] [-help]
```

Provide information about your servers as prompted or accept the defaults.

The deinstallation tool stops Oracle software, and removes Oracle software and configuration files on the operating system.

In addition, you can run the deinstallation tool with a parameter file, or select the following options to run the tool:

- `-home`

  Use this flag to indicate the home path of the Oracle home to check or deinstall. To deinstall Oracle software using the `deinstall` command in the Oracle home you plan to deinstall, provide a parameter file in another location, and do not use the `-home` flag.

  If you run `deinstall` from the `$ORACLE_HOME/deinstall` path, then the `-home` flag is not required because the tool knows from which home it is being run. If you use the standalone version of the tool or `runInstaller -deinstall` from the installation media, then `-home` is mandatory.

- `-silent`

  Use this flag to run the command in silent or response file mode. If you use the `-silent` flag, then you must use the `-paramfile` flag, and provide a parameter file that contains the configuration values for the Oracle home to deinstall or deconfigure.

  You can generate a parameter file to use or modify by running `deinstall` with the `-checkonly` flag. The `deinstall` command then discovers information from the Oracle home to deinstall and deconfigure. It generates the properties file, which you can then use with the `-silent` option.

  You can also modify the template file `deinstall.rsp.tmpl`, located in the `$ORACLE_HOME/deinstall/response` folder.

- `-checkonly`

  Use this flag to check the status of the Oracle software home configuration. Running the `deinstall` command with the `-checkonly` flag does not remove the Oracle configuration. The `-checkonly` flag generates a parameter file which you can then use with the `deinstall` command and `-silent` option.

- `-local`

  Use this flag on a multinode environment to deinstall Oracle software in a cluster.

  When you run `deinstall` with this flag, it deconfigures and deinstalls the Oracle software on the local node (the node where `deinstall` is run). On remote nodes, it deconfigures Oracle software, but does not deinstall the Oracle software.

- `-paramfile` *complete path of input parameter property file*

  Use this flag to run `deinstall` with a parameter file in a location other than the default. When you use this flag, provide the complete path where the parameter file is located.

  The default location of the parameter file depends on the location of `deinstall`:

  - From the installation media or stage location: `$ORACLE_HOME/inventory/response`

  - From a unzipped archive file from Oracle Technology Network: `/`*ziplocation*`/response`

- After installation from the installed Oracle home: `$ORACLE_HOME/deinstall/response`

- `-params [name1=value name 2=value name3=value ...]`

  Use this flag with a parameter file to override one or more values to change in a parameter file you have created.

- `-o complete path of directory for saving response files`

  Use this flag to provide a path other than the default location where the properties file (`deinstall.rsp.tmpl`) is saved.

  The default location of the parameter file depends on the location of `deinstall`:

  - From the installation media or stage location before installation: `$ORACLE_HOME/`

  - From a unzipped archive file from Oracle Technology Network: `/ziplocation/response/`

  - After installation from the installed Oracle home: `$ORACLE_HOME/deinstall/response`

- `-help`

  Use the help option (`-help`) to get additional information about the command option flags.

8.3.2 Example of Running the Deinstall Command for Oracle RAC

If you run the deinstallation tool using the deinstall command from the `$ORACLE_HOME/deinstall` folder, then the deinstallation starts without prompting you for an ORACLE_HOME.

Use the optional flag `-paramfile` to provide a path to a parameter file.

In the following example, the `runInstaller` command is in the path `/directory_path`, where `directory_path` is the path to the `database` directory on the installation media, and `/u01/app/oracle/product/12.1.0/dbhome_1/` is the path to the Oracle home to remove:

```
$ cd /directory_path/runInstaller
$ ./runInstaller -deinstall -home /u01/app/oracle/product/12.1.0/dbhome_1/
```

The following example uses a parameter file in the software owner location `/home/usr/oracle`:

```
$ cd /directory_path/runInstaller
$ ./runInstaller -deinstall -paramfile /home/usr/oracle/my_db_paramfile.tmpl
```

8.3.3 Deinstallation Parameter File Example for Oracle RAC

You can run the deinstallation tool on Oracle RAC Databases with the `-paramfile` option to use the values you specify in the parameter file. The following is an example of a parameter file, in which the Oracle Database binary owner is `oracle`, the Oracle Database home (Oracle home) is in the path `/u01/app/oracle/product/12.1.0/dbhome_1/`, the Oracle base (where other Oracle software is installed) is `/u01/app/oracle/`, the central Oracle Inventory home (oraInventory) is `/u01/app/oraInventory`, the virtual IP address (VIP) is `192.0.2.1`, the local node (the node where you run the deinstallation session from) is `myserver`, and the OSDBA group is `dba`:

```
#Copyright (c) 2005, 2006 Oracle Corporation. All rights reserved.
#Mon Feb 17 06:48:39 UTC 2014
DISK_GROUPS.sidb=
ASM_HOME=
ASM_LOCAL_SID=
LOGDIR=/u01/app/oracle/product/12.1.0/dbhome_1/oraInventory/logs/
ORACLE_BASE.sidb=/u01/app/oracle/
RECOVERY_LOC.sidb=
STORAGE_TYPE.sidb=FS
ORACLE_BASE=/u01/app/oracle/
INVENTORY_LOCATION=/u01/app/oraInventory
DB_TYPE.sidb=SI_DB
NODE_LIST.sidb=myserver
ARCHIVE_LOG_DESTINATION_LOC.sidb=
LOCAL_SID.sidb=sidb
DB_UNIQUE_NAME_LIST=sidb
ASM_FILES.sidb=
HOME_TYPE=SIDB
CRS_HOME=false
RAW_MAPPING_FILE.sidb=
SID_LIST.sidb=sidb
ORACLE_BINARY_OK=true
DATAFILE_LOC.sidb=/u01/app/oracle/oradata
local=false
LOCAL_NODE=myserver
CREATION_MODE.sidb=y
CONFIGFILE_LOC.sidb=
DIAG_DEST.sidb=/u01/app/oracle/
silent=false
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1/
SPFILE_LOC.sidb=
```

A

Troubleshooting Oracle Real Application Clusters Installations

This appendix provides troubleshooting information for installing Oracle Real Application Clusters (Oracle RAC). It contains the following sections:

- Troubleshooting Oracle Real Application Clusters Installations
- About Using CVU Cluster Healthchecks After Installation

> **See Also:** The Oracle Database 12*c* Release 1 (12.1) Oracle Real Application Clusters documentation set:
>
> - *Oracle Real Application Clusters Administration and Deployment Guide*
> - *Oracle Clusterware Administration and Deployment Guide*

A.1 Troubleshooting Oracle Real Application Clusters Installations

This section contains the following topics:

- General Installation Issues
- Oracle RAC Installation Error Messages
- Performing Cluster Diagnostics During Oracle Clusterware Installations
- Reviewing the Log of an Installation Session
- Configuration Assistant Errors

A.1.1 General Installation Issues

The following is a list of examples of the types of errors that can occur during installation. It contains the following issues:

- An error occurred while trying to get the disks
- Failed to connect to server, Connection refused by server, or Can't open display
- Nodes unavailable for selection from the OUI Node Selection screen
- Node nodename is unreachable
- ORA-00845: MEMORY_TARGET not supported on this system.
- PRKP-1001: Error starting instance
- Time stamp is in the future

■ YPBINDPROC_DOMAIN: Domain not bound

An error occurred while trying to get the disks

Cause: There is an entry in `/etc/oratab` pointing to a non-existent Oracle home. The OUI error file should show the following error: `"java.io.IOException: /home/oracle/OraHome//bin/kfod: not found"`

Action: Remove the entry in `/etc/oratab` pointing to a non-existing Oracle home.

Failed to connect to server, Connection refused by server, or Can't open display

Cause: These are typical of X Window display errors on Windows or UNIX systems, where xhost is not properly configured.

Action: In a local terminal window, log in as the user that started the X Window session, and enter the following command:

```
$ xhost fully_qualified_remote_host_name
```

For example:

```
$ xhost somehost.example.com
```

Then, enter the following commands, where *workstation_name* is the host name or IP address of your workstation.

Bourne, Bash, or Korn shell:

```
$ DISPLAY=workstation_name:0.0
$ export DISPLAY
```

To determine if X Window applications display correctly on the local system, enter the following command:

```
$ xclock
```

The X clock should appear on your monitor. If this fails, then use of the `xhost` command may be restricted on the server.

If you are using a VNC client to access the server, then ensure that you are accessing the visual that is assigned to the user that you are trying to use for the installation. For example, if you used the `su` command to become the installation owner on another user visual, and the `xhost` command use is restricted, then you cannot use the `xhost` command to change the display. If you use the visual assigned to the installation owner, then the correct display is available, and entering the `xclock` command displays the X clock.

Nodes unavailable for selection from the OUI Node Selection screen

Cause: Oracle Clusterware is either not installed, or the Oracle Clusterware services are not up and running.

Action: Install Oracle Clusterware, or review the status of your Oracle Clusterware. Consider restarting the nodes, as doing so may resolve the problem.

Node *nodename* is unreachable

Cause: Unavailable IP host

Action: Attempt the following:

1. Run the command `ifconfig -a`. Compare the output of this command with the contents of the `/etc/hosts` file to ensure that the node IP is listed.

2. Run the command `nslookup` to see if the host is reachable.

3. As the `oracle` user, attempt to connect to the node with `ssh` or `rsh`. If you are prompted for a password, then user equivalence is not set up properly. Contact your system administrator, or consult *Oracle Grid Infrastructure Installation Guide* for your platform to complete SSH configuration.

ORA-00845: MEMORY_TARGET not supported on this system.

Cause: Insufficient memory to support SGA and PGA sizes

Action: Ask your system administrator to increase the shared memory file system (for example, on Linux, increase `/dev/shm`).

PRKP-1001: Error starting instance

Cause: Missing ODBC Driver Manager. Associated message is **CRS-0215: Could not start resource**.

Action: Clean up installation, download and install the ODBC driver from http://www.unixodbc.org, and restart the installation. This is a requirement for Oracle RAC databases, documented in system requirements in *Oracle Grid Infrastructure Installation Guide* for your platform.

Time stamp is in the future

Cause: One or more nodes has a different clock time than the local node. If this is the case, then you may see output similar to the following:

```
time stamp 2005-04-04 14:49:49 is 106 s in the future
```

Action: Ensure that all member nodes of the cluster have the same clock time.

YPBINDPROC_DOMAIN: Domain not bound

Cause: This error can occur during postinstallation testing when a node public network interconnect is pulled out, and the VIP does not fail over. Instead, the node hangs, and users are unable to log in to the system. This error occurs when the Oracle home, `listener.ora`, Oracle log files, or any action scripts are located on an NAS device or NFS mount, and the name service cache daemon `nscd` has not been activated.

Action: Enter the following command on all nodes in the cluster to start the nscd service:

```
/sbin/service  nscd start
```

A.1.2 Oracle RAC Installation Error Messages

Note that the user performing the Oracle RAC installation must have membership both in the oinstall group and the OSDBA group (typically `oinstall` and `dba`). If this is not the case, then the installation will fail

A.1.3 Performing Cluster Diagnostics During Oracle Clusterware Installations

If Oracle Universal Installer (OUI) does not display the Node Selection page, then perform clusterware diagnostics by running the `olsnodes -v` command from the binary directory in your Oracle Clusterware home *(Grid_home*/bin on Linux and UNIX-based systems), and analyzing its output. Refer to your clusterware documentation if the detailed output indicates that your clusterware is not running.

In addition, use the following command syntax to check the integrity of the Cluster Manager:

```
cluvfy comp clumgr -n node_list -verbose
```

In the preceding syntax example, the variable *node_list* is the list of nodes in your cluster, separated by commas.

A.1.4 Reviewing the Log of an Installation Session

During an installation, Oracle Universal Installer (OUI) records all of the actions that it performs in a log file. If you encounter problems during the installation, then review the log file for information about possible causes of the problem.

To view the log file, follow these steps:

1. If necessary, enter the following command to determine the location of the oraInventory directory:

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

2. Enter the following command to determine the name of the log file:

   ```
   $ ls -ltr
   ```

   This command lists the files in the order of creation, with the most recent file shown last. Installer log files have names similar to the following, where *date_time* indicates the date and time that the installation started:

   ```
   installActionsdate_time.log
   ```

   To view the most recent entries in the log file, where information about a problem is most likely to appear, enter a command similar to the following:

   ```
   $ tail -50 installActions2007-07-20_09-53-22AM.log | more
   ```

   This command displays the last 50 lines in the log file, and enables you to page through them.

   If the error displayed by Oracle Universal Installer or listed in the log file indicates a relinking problem, then refer to the following file for more information:

   ```
   $ORACLE_HOME/install/make.log
   ```

A.1.5 Configuration Assistant Errors

To troubleshoot an installation error that occurs when a configuration assistant is running:

Review the installation log files listed in the section""Reviewing the Log of an Installation Session" on page A-4.

Review the specific configuration assistant log file located in the Oracle RAC installation owner Oracle base directory, in the path `$ORACLE_BASE/cfgtoollogs`. Try to fix the issue that caused the error.

If you see the "Fatal Error. Reinstall" message, then look for the cause of the problem by reviewing the log files. Refer to the section "Resolving Irrecoverable Errors Reported by Configuration Assistants" on page A-5 for further instructions.

This section contains the following topics:

- Configuration Assistant Failures

- Resolving Irrecoverable Errors Reported by Configuration Assistants

A.1.5.1 Configuration Assistant Failures

Oracle configuration assistant failures are noted at the bottom of the installation screen. The configuration assistant interface displays additional information, if available. The configuration assistant execution status is stored in the following file:

```
oraInventory_location/logs/installActionsdate_time.log
```

More details about errors related to the configuration assistant can be found in the following directory:

```
$ORACLE_BASE/cfgtoollogs
```

The Oracle base directory is the Oracle base for the Oracle RAC installation owner. Completion status codes are listed in the following table:

```
Status Result Code
Configuration assistant succeeded 0
Configuration assistant failed 1
Configuration assistant cancelled -1
```

A.1.5.2 Resolving Irrecoverable Errors Reported by Configuration Assistants

If you receive an irrecoverable (`fatal`) error while a configuration assistant is running, then you must complete the following tasks:

1. Remove the failed installation as described in Section A.3, "Cleaning Up After a Failed Installation".

2. Correct the cause of the irrecoverable error.

3. Reinstall the Oracle software.

> **See Also:** Chapter 8, "Removing Oracle Real Application Clusters Software"

A.2 About Using CVU Cluster Healthchecks After Installation

Starting with Oracle Grid Infrastructure 11*g* Release 2 (11.2.0.3) and later, you can use the CVU `healthcheck` command option to check your Oracle Clusterware and Oracle Database installations for their compliance with mandatory requirements and best practices guidelines, and to check to ensure that they are functioning properly.

Use the following syntax to run the `healthcheck` command option:

```
cluvfy comp healthcheck [-collect {cluster|database}] [-db db_unique_name]
[-bestpractice|-mandatory] [-deviations] [-html] [-save] [-savedir directory_path]
```

For example:

```
$ cd /home/grid/cvu_home/bin
$ ./cluvfy comp healthcheck -collect cluster -bestpractice -deviations -html
```

The options are:

- `-collect [cluster|database]`

  Use this flag to specify that you want to perform checks for Oracle Clusterware (`cluster`) or Oracle Database (`database`). If you do not use the `collect` flag with the `healthcheck` option, then `cluvfy comp healthcheck` performs checks for both Oracle Clusterware and Oracle Database.

- `-db db_unique_name`

Use this flag to specify checks on the database unique name that you enter after the `db` flag.

CVU uses JDBC to connect to the database as the user `cvusys` to verify various database parameters. For this reason, if you want checks to be performed for the database you specify with the `-db` flag, then you must first create the `cvusys` user on that database, and grant that user the CVU-specific role, `cvusapp`. You must also grant members of the `cvusapp` role `select` permissions on system tables.

A SQL script is included in *CVU_home*`/cv/admin/cvusys.sql` to facilitate the creation of this user. Use this SQL script to create the cvusys user on all the databases that you want to verify using CVU.

If you use the `db` flag but do not provide a database unique name, then CVU discovers all the Oracle Databases on the cluster. To perform best practices checks on these databases, you must create the `cvusys` user on each database, and grant that user the `cvusapp` role with the `select` privileges needed to perform the best practice checks.

- `[-bestpractice | -mandatory] [-deviations]`

  Use the `bestpractice` flag to specify best practice checks, and the `mandatory` flag to specify mandatory checks. Add the `deviations` flag to specify that you want to see only the deviations from either the best practice recommendations or the mandatory requirements. You can specify either the `-bestpractice` or `-mandatory` flag, but not both flags. If you specify neither `-bestpractice` or `-mandatory`, then both best practices and mandatory requirements are displayed.

- `-html`

  Use the `html` flag to generate a detailed report in HTML format.

  If you specify the `html` flag, and a browser CVU recognizes is available on the system, then the browser is started and the report is displayed on the browser when the checks are complete.

  If you do not specify the `html` flag, then the detailed report is generated in a text file.

- `-save [-savedir dir_path]`

  Use the `save` or `-save -savedir` flags to save validation reports (`cvuchecdkreport_`*timestamp*`.txt` and `cvucheckreport_`*timestamp*`.htm`), where *timestamp* is the time and date of the validation report.

  If you use the `save` flag by itself, then the reports are saved in the path *CVU_home*`/cv/report`, where *CVU_home* is the location of the CVU binaries.

  If you use the flags `-save -savedir`, and enter a path where you want the CVU reports saved, then the CVU reports are saved in the path you specify.

A.3 Cleaning Up After a Failed Installation

If an installation fails, then you must remove the Oracle home directory and remove all files that OUI created during the attempted installation. Perform the following steps to clean up the failed installation:

1. Follow the instructions in Chapter 8, "Removing Oracle Real Application Clusters Software" to run OUI to deinstall Oracle RAC.

2. Manually remove the directory that was used as the Oracle home directory during the installation.

After you have completed these steps, you can start the installation again.

B

Using Scripts or Response Files to Create Oracle RAC Databases

This appendix describes the steps for noninteractive installations during which you can create Oracle Real Application Clusters (Oracle RAC) databases using scripts.

> **Note:** The scripts generated by Database Configuration Assistant (DBCA) are for reference only. Oracle strongly recommends that you use DBCA to create a database.

> **See Also:** *Oracle Universal Installer NextGen Installation Guide* for detailed information about generating scripts

This appendix contains the following topics:

- Using DBCA to Generate Installation Scripts for Oracle RAC
- About DBCA Noninteractive (Silent) Configuration for Oracle RAC
- DBCA Commands for Noninteractive (Silent) Configuration for Oracle RAC
- How Response Files Work
- Preparing Response Files
- Running Oracle Universal Installer Using a Response File
- Running Net Configuration Assistant Using Response Files
- Running Configuration Assistants Using Response Files
- Postinstallation Configuration Using Response Files

B.1 Using DBCA to Generate Installation Scripts for Oracle RAC

To generate scripts to create an Oracle RAC database, create a database using the generated scripts, and prepare the database for use, complete the following steps:

1. Start DBCA and select your preferred options to build the Oracle RAC database.

    On the Creation Options page of your DBCA session, deselect Create Database and select **Generate Database Creation Scripts** before you click **Finish**. You can accept the default destination directory for the scripts, or browse for a different location. In either case, you should note the path name for use in the next step.

> **See Also:** Chapter 3, "Creating Oracle RAC or Oracle RAC One Node Databases with DBCA" for additional information about running a DBCA session

2. Go to the directory where DBCA created the scripts, and review the SQL scripts to ensure that they contain the statements to build a database with the characteristics you require. If they do not, then Oracle recommends that you rerun DBCA to create scripts with the desired configuration rather than editing the scripts yourself.

3. On each cluster node you identified during your DBCA session, run the script `sid.sh`, where `sid` is the SID prefix that you entered on the DBCA Database Name page.

4. Set the initialization parameter, `cluster_database`, to the value `TRUE` in your SPFILE by running an `ALTER SYSTEM` statement in SQL*Plus, or by uncommenting the parameter in the PFILE for each instance.

5. Configure Oracle Net Services to support your new database and instances as described in Chapter 7, "Understanding the Oracle RAC Installed Configuration".

6. Set the REMOTE_LISTENER parameter to the SCAN (using the Easy Connect Naming syntax *scanname*:*scanport*) in your SPFILE by entering an ALTER SYSTEM statement in SQL*Plus, or by uncommenting the parameter in the PFILE for each instance.

7. Run the Server Control Utility (SRVCTL) to configure and start database and instance applications as described in *Oracle Real Application Clusters Administration and Deployment Guide*.

> **See Also:** The *Oracle Universal Installer NextGen Installation Guide* for additional information about creating and using scripts to install Oracle software with Oracle Universal Installer (OUI)

B.2 About DBCA Noninteractive (Silent) Configuration for Oracle RAC

You can perform a noninteractive, or *silent* configuration of Oracle RAC using DBCA. To perform a silent configuration, you must have completed an Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) installation, run the `root.sh` script from the Oracle Database home, and defined the Oracle home directory environment variable.

You can use DBCA to create a database from templates supplied by Oracle, or from templates that you create. The templates contain settings optimized for a particular type of workload.

Oracle provides templates for the following two workload types:

- General purpose or transaction processing

- Data warehouse

For more complex environments, you can select the Custom Database option. This option does not use templates and results in a more extensive installation interview, which means that it takes longer to create your database.

B.3 DBCA Commands for Noninteractive (Silent) Configuration for Oracle RAC

You can use the following command syntax to create an Oracle RAC database using the general purpose template, placing the data files in an existing Oracle ASM disk group. Nodes `node1` and `node2` are the cluster nodes on which Oracle RAC database instances are created. The disk group name is `+ASMgrp1`, and *password* is a placeholder for a password. The passwords can be all the same password or different passwords each time:

```
# su oracle -c "$ORACLE_HOME/bin/dbca -silent -createDatabase -templateName
General_Purpose.dbc -gdbName $DBNAME -sid $ORACLE_SID -sysPassword password
-systemPassword password -sysmanPassword password -dbsnmpPassword password
-emConfiguration LOCAL -storageType ASM -diskGroupName ASMgrp1
-datafileJarLocation $ORACLE_HOME/assistants/dbca/templates -nodeinfo
node1,node2 -characterset WE8ISO8859P1 -obfuscatedPasswords false -sampleSchema
false -asmSysPassword password"
```

> **See Also:** *Oracle Database Administrator's Guide* for a complete description of the dbca commands and options

B.4 How Response Files Work

When you start Oracle Universal Installer (OUI), you can use a response file to automate the installation and configuration of Oracle software, either fully or partially. OUI uses the values contained in the response file to provide answers to some or all installation prompts.

Typically, the installer runs in interactive mode, which means that it prompts you to provide information in graphical user interface (GUI) screens. When you use response files to provide this information, you run the installer from a command prompt using either of the following modes:

- Silent mode

  If you include responses for all of the prompts in the response file and specify the `-silent` option when starting the installer, then it runs in silent mode. During a silent mode installation, the installer does not display any screens. Instead, it displays progress information in the terminal that you used to start it.

- Response file mode

  If you include responses for some or all of the prompts in the response file and omit the `-silent` option, then the installer runs in response file mode. During a response file mode installation, the installer displays all the screens, screens for which you specify information in the response file, and also screens for which you did not specify the required information in the response file.

You define the settings for a silent or response file installation by entering values for the variables listed in the response file. For example, to specify the Oracle home name, supply the appropriate value for the `ORACLE_HOME` environment variable:

```
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
```

Another way of specifying the response file variable settings is to pass them as command-line arguments when you run the installer. For example:

```
-silent directory_path
```

In this command, *directory_path* is the path of the `database` directory on the DVD or the path of the directory on the hard drive.

> **See Also:** *Oracle Universal Installer NextGen Installation Guide* for more information about response files

B.4.1 Reasons for Using Silent Mode or Response File Mode

The following table provides use cases for running the installer in silent mode or response file mode.

Mode	Uses
Silent	Use silent mode for the following installations:
	■ Complete an unattended installation, which you schedule using operating system utilities such as `at`.
	■ Complete several similar installations on multiple systems without user interaction.
	■ Install the software on a system that does not have X Window System software installed on it.
	The installer displays progress information on the terminal that you used to start it, but it does not display any of the installer screens.
Response file	Use response file mode to complete similar Oracle software installations on more than one system, providing default answers to some, but not all of the installer prompts.
	If you do not specify information required for a particular installer screen in the response file, then the installer displays that screen. It suppresses screens for which you have provided all of the required information.

B.4.2 Creating a Database Using Oracle ASM for Database Files

Before you create a database that uses Oracle Automatic Storage Management (Oracle ASM), you must run the `root.sh` script. For this reason, you cannot create a database using Oracle ASM as the storage option for database files during a silent mode installation. Instead, you can complete a software-only installation using silent-mode, and then run the Oracle Net Configuration Assistant and DBCA in silent mode after you have completed the software-only installation and you have run the `root.sh` script.

> **Note:** This limitation applies only to databases that use Oracle Automatic Storage Management as the storage option for database files. You can create a database that uses the file system option during a silent-mode installation.

B.4.3 General Procedure for Using Response Files

Use the following general steps to install and configure Oracle products using the installer in silent or response file mode:

> **Note:** You must complete all required preinstallation tasks on a system before running the installer in silent or response file mode.

1. Create the `oraInst.loc` file if it is not present on the server.

2. Prepare a response file.

3. Run the installer in silent or response file mode.

4. If you completed a software-only installation, then run Net Configuration Assistant and DBCA in silent or response file mode.

These steps are described in the following sections.

B.5 Preparing Response Files

This section describes the following methods to prepare response files for use during silent mode or response file mode installations:

- Editing a Response File Template
- Recording Response Files

B.5.1 Editing a Response File Template

For Oracle Database, these files are located in the database/response directory on the installation media or in the *Oracle_home*/inventory/response directory after the software has been installed.

> **Note:** If you copied the software to a hard disk, then the response files are located in the directory /response.

Table B–1 lists the response files provided with this software:

Table B–1   Response Files for Oracle Database

Response File	Description
db_install.rsp	Silent installation of Oracle Real Application Clusters
dbca.rsp	Silent creation and configuration of Oracle Database using DBCA
netca.rsp	Silent configuration of Oracle Net using NETCA

> **Caution:** When you modify a response file template and save a file for use, the response file may contain plain text passwords. Ownership of the response file should be given to the Oracle software installation owner only, and permissions on the response file should be changed to 600. Oracle strongly recommends that database administrators or other administrators delete or secure response files when they are not in use.

To copy and modify a response file:

1. Copy the response file from the response file directory to a directory on your system:

```
$ cp /directory_path/response/response_file.rsp local_directory
```

In this example, *directory_path* is the path to the database directory on the installation media. If you have copied the software to a hard drive, then you can edit the file in the response directory if you prefer.

2. Open the response file in a text editor:

```
$ vi /local_dir/response_file.rsp
```

> **See Also:** The *Oracle Universal Installer NextGen Installation Guide*for detailed information on creating response files

3. Follow the instructions in the file to edit it.

> **Note:** The installer or configuration assistant fails if you do not correctly configure the response file.

4. Secure the response file by changing the permissions on the file to 600:

```
$ chmod 600 /local_dir/response_file.rsp
```

Ensure that only the Oracle software owner user can view or modify response files or consider deleting them after the installation succeeds.

> **Note:** A fully-specified response file for an Oracle Database installation contains the passwords for database administrative accounts and for a user who is a member of the OSDBA group (required for automated backups).

B.5.2 Recording Response Files

You can use OUI in interactive mode to record response files, which you can then edit and use to complete silent mode or response file mode installations. This method is useful for Advanced or software-only installations.

You can save all the installation steps into a response file during installation by clicking **Save Response File** on the Summary page. You can use the generated response file for a silent installation later.

When you record the response file, you can either complete the installation, or you can exit from the installer on the Summary page, before OUI starts to copy the software to the system.

If you use record mode during a response file mode installation, then the installer records the variable values that were specified in the original source response file into the new response file.

> **Note:** You cannot save passwords while recording the response file.

To record a response file:

1. Complete preinstallation tasks as for a standard installation.

When you run the installer to record a response file, it checks the system to verify that it meets the requirements to install the software. For this reason, Oracle recommends that you complete all of the required preinstallation tasks and record the response file while completing an installation.

2. Ensure that the Oracle software owner user (typically `oracle`) has permissions to create or write to the Oracle home path that you will specify when you run the installer.

3. On each installation screen, specify the required information.

4. When the installer displays the Summary screen, perform the following steps:

    a. Click **Save Response File**. In the window, specify a file name and location for the new response file. Click Save to write the responses you entered to the response file.

    b. Click **Finish** to continue with the installation.

       Click **Cancel** if you do not want to continue with the installation. The installation stops, but the recorded response file is retained.

5. If you do not complete the installation, then delete the Oracle home directory that the installer created using the path you specified in the Specify File Locations screen.

6. Before you use the saved response file on another system, edit the file and make any required changes. Use the instructions in the file as a guide when editing it.

B.6 Running Oracle Universal Installer Using a Response File

Now, you are ready to run OUI at the command line, specifying the response file you created, to perform the installation.

B.6.1 Viewing the Help Information for OUI

For information about all the available options for OUI, run the `runInstaller` command with the `-help` option, for example:

```
$ directory_path/runInstaller -help
```

The help information appears in your session window after a brief delay.

B.6.2 Running OUI Using a Response File for Input

To run the installer with a response file to install the Oracle Database software, complete the following steps:

1. Complete the preinstallation tasks as for a typical installation.

2. Log in as the software installation owner user.

3. If you are completing a response file mode installation, set the `DISPLAY` environment variable.

    > **Note:** You do not have to set the `DISPLAY` environment variable if you are completing a silent mode installation.

4. To start the installer in silent or response file mode, enter a command similar to the following:

```
$ /directory_path/runInstaller [-silent] [-noconfig] \
 -responseFile responsefilename
```

> **Note:** Do not specify a relative path for the response file. If you specify a relative path, then the installer fails.

In this example:

- *directory_path* is the path of the `database` directory on the DVD or the path of the directory on the hard drive.

- `-silent` runs the installer in silent mode.

- `-noconfig` suppresses running the configuration assistants during installation, and a software-only installation is performed instead.

- *responsefilename* is the full path and file name of the installation response file that you configured previously. See "Editing a Response File Template" on page B-5.

> **Note:** For more information about other options for the `runInstaller` command, enter the following command:
>
> ```
> $ /directory_path/runInstaller -help
> ```

5. When the installation completes, log in as the `root` user and run the `root.sh` script:

```
$ sudo sh
password:
# /oracle_home_path/root.sh
```

B.7 Postinstallation Configuration Using Response Files

The following sections explain how to create and run a response file configuration after installing Oracle software. The topics in this section are:

- About the Postinstallation Configuration File

- Creating a Password Response File

- Running Postinstallation Configuration Using a Response File

B.7.1 About the Postinstallation Configuration File

When you run a silent or response file installation, you provide information about your servers in a response file that you otherwise provide manually during a graphical user interface installation. However, the response file does not contain passwords for user accounts that configuration assistants require after software installation is complete. The configuration assistants are started with a script called `configToolAllCommands`. You can run this script in response file mode by using a password response file. The script uses the passwords to run the configuration tools in succession to complete configuration.

If you keep the password file to use for clone installations, then Oracle strongly recommends that you store the password file in a secure location. In addition, if you have to stop an installation to fix an error, then you can run the configuration assistants using `configToolAllCommands` and a password response file.

The `configToolAllCommands` password response file has the following syntax options:

- *internal_component_name* is the name of the component that the configuration assistant configures

- *variable_name* is the name of the configuration file variable

- *value* is the desired value to use for configuration.

The command syntax is as follows:

internal_component_name|variable_name=value

For example:

```
oracle.assistants.asm|S_ASMPASSWORD=myPassWord
```

Oracle Database configuration assistants require the SYS, SYSTEM, and DBSNMP passwords for use with DBCA. You might need to specify the following additional passwords, depending on your system configuration:

- If the database is using Oracle ASM for storage, then you must specify a password for the S_ASMSNMPPASSWORD variable. If you are not using Oracle ASM, then leave the value for this password variable blank.

- If you create a multitenant container database (CDB) with one or more pluggable databases (PDBs), then you must specify a password for the S_PDBADMINPASSWORD variable. If you are not using Oracle ASM, then leave the value for this password variable blank.

Oracle strongly recommends that you maintain security with a password response file:

- Permissions on the response file should be set to 600.

- The owner of the response file should be the installation owner user, with the group set to the central inventory (oraInventory) group.

B.7.2 Creating a Password Response File

To create a password response file to use with the configuration assistants, perform the following steps:

1. Create a response file that has a name of the format *filename*.properties, for example:

   ```
   $ touch pwdrsp.properties
   ```

2. Open the file with a text editor, and cut and paste the sample password file contents, as shown in Example B–1, modifying as needed.

3. Change permissions to secure the password response file. For example:

   ```
   $ ls -al pwdrsp.properties
   -rw------- 1 oracle oinstall 0 Apr 30 17:30 pwdrsp.properties
   ```

Example B–1   Password response file for Oracle Real Application Clusters

This example provides a template for a password response file to use with the database configuration assistants.

```
oracle.assistants.server|S_SYSPASSWORD=password
oracle.assistants.server|S_SYSTEMPASSWORD=password
oracle.assistants.server|S_SYSMANPASSWORD=password
oracle.assistants.server|S_DBSNMPPASSWORD=password
oracle.assistants.server|S_ASMSNMPPASSWORD=password
oracle.assistants.server|S_PDBADMINPASSWORD=password
```

B.7.3 Running Postinstallation Configuration Using a Response File

To run configuration assistants with the `configToolAllCommands` script:

1. Create a password response file as described in "Creating a Password Response File" on page B-9.

2. Change directory to `$ORACLE_HOME/cfgtoollogs`.

3. Run the configuration script using the following syntax:

   ```
   configToolAllCommands RESPONSE_FILE=/path/name.properties
   ```

   For example:

   ```
   $ ./configToolAllCommands RESPONSE_FILE=/home/oracle/pwdrsp.properties
   ```

B.8 Running Configuration Assistants Using Response Files

You can run configuration assistants in response file or silent mode to configure and start Oracle software after it is installed on the system. To run configuration assistants in response file or silent mode, you must copy and edit a response file template.

> **Note:** If you copied the software to a hard disk, then the response file template is located in the `/response` directory.

This section contains the following topics:

- Running Database Configuration Assistant in Response File or Silent Mode
- Running Net Configuration Assistant Using Response Files

B.8.1 Running Database Configuration Assistant in Response File or Silent Mode

You can run Database Configuration Assistant (DBCA) in response file mode, either in silent mode or progress only mode, to configure and start an Oracle database on the system. To run Database Configuration Assistant in response file mode, you must copy and edit a response file template. Oracle provides a response file template named `dbca.rsp` in the `ORACLE_HOME/assistants/dbca` directory and also in the `/response` directory on the DVD.

To run DBCA in response file mode, you must use the `-responseFile` flag in combination with either the `-silent` or `-progressOnly` flag. You must also use a graphical display and set the `DISPLAY` environment variable.

To run Database Configuration Assistant in response file mode:

1. Copy the `dbca.rsp` response file template from the response file directory to a directory on your system:

   ```
   $ cp /directory_path/response/dbca.rsp local_directory
   ```

   In this example, `directory_path` is the path of the `database` directory on the DVD. If you have copied the software to a hard drive, you can edit the file in the `response` directory if you prefer.

> **Note:** As an alternative to editing the response file template, you can
> also create a database by specifying all required information as
> command line options when you run DBCA. For information about
> the list of options supported, enter the following command:
>
> ```
> $ $ORACLE_HOME/bin/dbca -help
> ```

2. Open the response file in a text editor:

   ```
   $ vi /local_dir/dbca.rsp
   ```

3. Edit the file, following the instructions in the file.

   > **Note:** DBCA fails if you do not correctly configure the response file.

4. Log in as the Oracle software owner user, and set the ORACLE_HOME environment variable to specify the correct Oracle home directory.

5. To run DBCA in response file mode, set the DISPLAY environment variable.

6. Use the following command syntax to run DBCA in silent or response file mode using a response file:

   ```
   $ORACLE_HOME/bin/dbca {-progressOnly | -silent} -responseFile \
   /local_dir/dbca.rsp
   ```

   In this example:

   - The -silent option indicates that DBCA should run in silent mode.

   - The -progressOnly option indicates that DBCA should run in response file mode.

   - *local_dir* is the full path of the directory where you copied the dbca.rsp response file template.

B.8.2 Running Net Configuration Assistant Using Response Files

You can run Net Configuration Assistant in silent mode to configure and start an Oracle Net listener on the system, configure naming methods, and configure Oracle Net service names. To run Net Configuration Assistant in silent mode, you must copy and edit a response file template. Oracle provides a response file template named netca.rsp in the response directory in the database/response directory on the DVD.

> **Note:** If you copied the software to a hard disk, then the response file template is located in the database/response directory.

To run Net Configuration Assistant using a response file:

1. Copy the netca.rsp response file template from the response file directory to a directory on your system:

   ```
   $ cp /directory_path/response/netca.rsp local_directory
   ```

   In this example, *directory_path* is the path of the database directory on the DVD. If you have copied the software to a hard drive, you can edit the file in the response directory if you prefer.

2. Open the response file in a text editor:

```
$ vi /local_dir/netca.rsp
```

3. Follow the instructions in the file to edit it.

> **Note:** Net Configuration Assistant fails if you do not correctly configure the response file.

4. Log in as the Oracle software owner user, and set the ORACLE_HOME environment variable to specify the correct Oracle home directory.

5. Enter a command similar to the following to run Net Configuration Assistant in silent mode:

```
$ $ORACLE_HOME/bin/netca -silent -responsefile /local_dir/netca.rsp
```

In this command:

- The -silent option indicates to run Net Configuration Assistant in silent mode.

- *local_dir* is the full path of the directory where you copied the netca.rsp response file template.

C

Directory Structure for Oracle RAC Environments

This appendix describes the directory structures for Oracle Real Application Clusters (Oracle RAC) software environments. This appendix contains the following topics:

- Understanding the Oracle RAC Directory Structure
- Directory Structures for Oracle RAC

C.1 Understanding the Oracle RAC Directory Structure

When you install Oracle Database 12*c* Release 1 (12.1) with Oracle RAC, all subdirectories except for the Oracle Inventory directory, the Oracle Automatic Storage Management home (if applicable), and the Oracle Clusterware home, are under a top-level Oracle base directory. The Oracle home and `admin` directories are also located under the Oracle base directory.

> **See Also:** *Oracle Database Installation Guide* for your platform for more information about the Oracle home and `admin` directories

C.2 Directory Structures for Oracle RAC

Table C–1 shows the hierarchical directory tree of a sample Optimal Flexible Architecture (OFA)-compliant database for Oracle RAC:

Table C–1   Directory Structure for A Sample OFA-Compliant Environment

Root	Second-level	Third-level	Fourth-level	Description
$ORACLE_BASE				`/u01/app/oracle` The default `ORACLE_BASE` directory, where the software owner is the `oracle` user.
	installation_type			`/u01/app/oracle/product/12.1.0/db` Type of installation under the Oracle base directory. For example, when installing Oracle Database 12c Release 1 (12.1), the value for installation type is `product/12.1.0/db`.
		$ORACLE_HOME		The location in which the Oracle Database software is installed. You can also choose to add a counter, if you have multiple installations of the software. For example, if you are creating a second Oracle home for the Oracle Database 12c Release 1 (12.1) software, then the path is as follows: `/u01/app/oracle/12.1.0/db_home2` Note that the Oracle database home is under the Oracle base directory path. An Oracle Clusterware directory must *not* be under the Oracle base directory path where the database executable files are located.
			dbs	The directory in which the local initialization parameter file is stored for the database.
	admin			`/u01/app/oracle/admin` The administrative directory. Note that with Oracle Database 11*g*, `bdump`, `cdump`, and `udump` files are relocated to the directory associated with `ADR_BASE`.
		db_unique_name		The database unique name; this is the same as `dbname` when the database name is 8 or fewer characters in length. For example, if your database name is `sales`, the directory path is: `/u01/app/oracle/admin/sales`
			/hdump /pfile	The dump destinations for database server.
$ADR_BASE			/bdump /cdump /udump	Automatic Diagnostic Repository dump destination trace files. Note that this directory path is set by the initialization parameter `DIAGNOSTIC_DEST`, and that the path for the Automatic Data Repository must be located on a shared storage location available to all the nodes. By default, this path is a subset of the Oracle base directory, in the following path: `$ORACLE_BASE/diag/`
Oracle Grid Infrastructure for a cluster home (Grid home)				An OFA-compliant path for the Oracle Clusterware home. The default value is: `/u01/app/12.1.0/grid` During the Oracle Grid Infrastructure for a cluster installation, Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM) software is installed. The `root.sh` script changes permissions of all of the parent directories of the Oracle Clusterware home directory to grant write access only to the root user. Because of this, the Oracle Clusterware home directory must not be a subdirectory of the Oracle base directory.
	bin			The subtree for Oracle Clusterware and Oracle ASM executable files.
	network			The subtree for Oracle Net Services configuration files and utilities.

> **See Also:** *Oracle Database Installation Guide* for your platform for more information about the Optimal Flexible Architecture standard

D

How to Stop Processes in an Existing Oracle RAC Database

This appendix describes how to stop all processes in an Oracle Real Application Clusters (Oracle RAC) database, in preparation for adding additional products to an existing database, or in preparation for patch updates. This appendix contains the following topics:

- Backing Up the Oracle RAC Database
- Shutting Down Oracle RAC Databases
- Stopping All Oracle Processes
- Using CVU to Validate Readiness for Oracle RAC Upgrades

D.1 Backing Up the Oracle RAC Database

Before you make any changes to the Oracle software, Oracle recommends that you create a backup of the Oracle Database installation.

D.2 Shutting Down Oracle RAC Databases

Shut down any existing Oracle Database instances on each node, with normal or immediate priority. You can use Oracle Enterprise Manager or the Server Control (SRVCTL) utility to shut down the Oracle RAC databases and their instances.

> **Note:** If you are using Oracle Clusterware or Oracle Restart, then you must shut down all Oracle Database instances on all cluster nodes before modifying the Oracle software. If you are performing a patch update, then review the instructions in the Patch Set Notes for detailed instructions.

See Also:

Oracle Database Administrator's Guide for more information about Oracle Restart

D.3 Stopping All Oracle Processes

Stop all listener and other processes running in the Oracle home directories where you want to modify the database software.

> **Note:** Before you can shut down Oracle ASM instances, Oracle Clusterware, or Oracle Restart, you must first shut down all database instances that use them.

D.3.1 Stopping Oracle Database Processes Before Adding Products or Upgrading

This section provides an overview of what must be done before adding additional products to Oracle Database 12*c* Release 1 (12.1). If you are updating only the Oracle Database software, then you only need to stop the databases instances that run from the Oracle home being updated. If you are performing a patch upgrade, then refer to the Database Patch Set Notes for the patch for additional instructions.

> **Note:** You must perform these steps in the order listed.

1. Shut down any processes in the Oracle home on each node that can access a database.

   > **Note:** Before you shut down any processes that are monitored by Oracle Enterprise Manager Cloud Control, set a blackout in Oracle Enterprise Manager Cloud Control for the processes that you intend to shut down. This is necessary so that the availability records for these processes indicate that the shutdown was planned downtime, rather than an unplanned system outage.

2. Shut down all Oracle RAC instances on all nodes. To shut down all Oracle RAC instances for a database, enter the following command, where *db_name* is the name of the database:

   ```
   $ oracle_home/bin/srvctl stop database -db db_name
   ```

D.3.2 Stopping Oracle Clusterware and Oracle ASM Instances

If you are upgrading the Oracle Clusterware or Oracle ASM software, then you must first stop all Oracle RAC database instances and all Oracle database instances that use Oracle ASM for storage. After stopping the database instances, shut down the Oracle ASM instances and Oracle Clusterware, as described in the following steps:

> **Note:** You must perform these steps in the order listed.

1. Shut down any processes in the Oracle home on each node that might be accessing a database, for example, the Oracle Enterprise Manager Cloud Control agent or an Oracle Net Services listener.

   > **Note:** Before you shut down any processes that are monitored by Oracle Enterprise Manager Cloud Control, set a blackout in Oracle Enterprise Manager Cloud Control for the processes that you intend to shut down. This is necessary so that the availability records for these processes indicate that the shutdown was planned downtime, rather than an unplanned system outage.

2. Shut down all Oracle RAC instances on all nodes. To shut down all Oracle RAC instances for a database, enter the following command, where *db_name* is the name of the database:

```
$ oracle_home/bin/srvctl stop database -db db_name
```

3. Stop all Oracle Clusterware node applications node applications and resources on all nodes. To stop node applications running on a node, enter the following command, where `u01/app/grid/12.1.0` is the Grid home, and *node* is the name of the node where the applications run:

```
$ u01/app/grid/12.1.0/bin/srvctl stop nodeapps -node node
```

4. Shut down the Oracle Clusterware processes and the Oracle ASM instance on each node by entering the following command on all nodes as the `root` user:

```
# crsctl stop cluster
```

To stop Oracle Clusterware and Oracle ASM instances on all nodes in the cluster, you can enter a single command: `crsctl stop cluster -all`.

> **Note:** The `crsctl stop crs` command is used to stop Oracle Restart services for single-instance databases.

D.4 Using CVU to Validate Readiness for Oracle RAC Upgrades

Review the contents in this section to validate that your Oracle RAC cluster is ready for upgrades.

D.4.1 About the CVU Database Upgrade Validation Command Options

Use the Cluster Verification Utility (CVU) command `cluvfy stage -predbinst -upgrade` to check the readiness of your Oracle RAC installation for upgrades. Running `cluvfy` with the `-predbinst -upgrade` flags performs system checks to confirm if the cluster is in a correct state for upgrading from an existing Oracle RAC installation.

The command uses the following syntax, where variable content is indicated by italics:

```
cluvfy stage -pre dbinst -upgrade -src_dbhome src_RAChome  -dest_dbhome
dest_RAChome -dest_version dest_version [-fixup[-fixupdirpath]] [-verbose]
```

The options are:

- `-src_dbhome` *src_RAChome*

  Use this flag to indicate the location of the source Oracle RAC home that you are upgrading, where *src_RAChome* is the path to the home to upgrade.

- `-dest_dbhome` *dest_RAChome*

  Use this flag to indicate the location of the upgrade Oracle RAC home, where *dest_RAChome* is the path to the Oracle RAC home.

- `-dest_version` *dest_version*

  Use this flag to indicate the release number of the upgrade, including any patchset. The release number must include the five digits designating the release to the level of the platform-specific patch. For example: 12.1.0.1.0.

> **See Also:** *Oracle Database Administrator's Guide* for information about release number format

- `-fixup` [`-fixupdir` *fixupdirpath*]]

  Use the `-fixup` flag to indicate that you want to generate instructions for any required steps you must complete to ensure that your cluster is ready for an upgrade. The default location is the CVU work directory. To place the fixup instructions in a different directory, add the flag `-fixupdir`, and provide the path to the fixup directory.

- `-verbose`

  Use the `-verbose` flag to produce detailed output of individual checks.

D.4.2 Example of Verifying System Upgrade Readiness for Oracle RAC Infrastructure

You can verify that the permissions required for installing Oracle RAC have been configured on the nodes `node1` and `node2` by running the following command:

```
$ ./cluvfy stage -pre dbinst -upgrade -src_dbhome
/u01/app/oracle/product/11.2.0/dbhome_1 -dest_dbhome
/u01/app/oracle/product/12.1.0/dbhome_1 -dest_version 12.1.0.0.0 -fixup
-fixupdirpath /home/oracle/fixup -verbose
```

D.4.3 Verifying System Readiness for Oracle Database Upgrades

Use Cluster Verification Utility (CVU) to assist you with system checks in preparation for starting a database upgrade. The installer runs the appropriate CVU checks automatically, and either prompts you to fix problems, or provides a fixup script to be run on all nodes in the cluster before proceeding with the upgrade.

> **See Also:** *Oracle Database Upgrade Guide*

E

Managing Oracle Database Port Numbers

This appendix lists the default port numbers and describes how to change the assigned port after installation. This appendix contains the following topics:

- About Managing Ports
- Viewing Port Numbers and Access URLS
- Port Numbers and Protocols of Oracle Components

E.1 About Managing Ports

During installation, Oracle Universal Installer (OUI) assigns port numbers to components from a set of default port numbers. Many Oracle Real Application Clusters (Oracle RAC) components and services use ports. As an administrator, it is important to know the port numbers used by these services, and to ensure that the same port number is not used by two services on your system.

Most port numbers are assigned during installation. Every component and service has an allotted port range, which is the set of port numbers Oracle RAC attempts to use when assigning a port. Oracle RAC starts with the lowest number in the range and performs the following checks:

- Is the port used by another Oracle Database installation on the system?

  The installation can be either active or inactive at the time; Oracle Database can still detect if the port is used.

- Is the port used by a process that is currently running?

  This could be any process on the host, including processes other than Oracle Database processes.

If the answer to any of the preceding questions is yes, then Oracle RAC moves to the next highest port in the allotted port range and continues checking until it finds a free port.

E.2 Viewing Port Numbers and Access URLS

In most cases, the Oracle Database component's port number is listed in the tool used to configure the port. In addition, ports for some Oracle Database applications are listed in the `portlist.ini` file. This file is located in the directory *$ORACLE_HOME*/install.

If you change a port number after installation, then it is not updated in the `portlist.ini` file, so you can rely on this file only immediately after installation. To find or change a port number, use the methods described in this appendix.

E.3 Port Numbers and Protocols of Oracle Components

Table E–1 lists the port numbers and protocols used by components that are configured during the installation. By default, the first port in the range is assigned to the component, if it is available.

Table E–1   Ports Used in Oracle Components

Component and Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
Cluster Manager The port number is assigned automatically during installation. You cannot view or modify it afterward.	Dynamic	Dynamic	UDP	Yes
Cluster Synchronization Service daemon (CSSD) The Cluster Synchronization Service (CSS) daemon uses a fixed port for node restart advisory messages. This port is used on all interfaces that have broadcast capability. Broadcast occurs only when a node eviction restart is imminent.	42424	Dynamic	UDP	Yes
Grid Plug and Play (GPNPD) GPNPD provides access to the Grid Plug and Play profile, and coordinates updates to the profile among the nodes of the cluster to ensure that all of the nodes have the most recent profile.	Dynamic	Dynamic	TCP	No
Multicast Domain Name Service (MDNSD) The mDNS process is a background process on Linux and UNIX, and a service on Window, and is necessary for Grid Plug and Play and GNS.	5353	Dynamic	UDP/TCP	No
Oracle Cluster Registry The port number is assigned automatically during installation. You cannot view or modify it afterward.	Dynamic	Dynamic	UDP	Yes
Oracle Clusterware Daemon (CRSD) Oracle Clusterware daemon internode connection. The port number is assigned automatically during installation. You cannot view or modify it afterward.	Dynamic	Dynamic	UDP	Yes
Oracle Connection Manager Listening port for Oracle client connections to Oracle Connection Manager. You can configure Oracle Connection Manager after installation using NETCA.	1630	1630	TCP	No
Oracle Containers for J2EE (OC4J) The CRS Agent uses port 8888 locally to manage the lifecycle of the container.	8888	8888	TCP	Not applicable
Oracle Data Guard Shares the Oracle Net listener port and is configured during installation. To reconfigure this port, use Oracle Net Configuration Assistant (NETCA) to reconfigure the listener.	1521 (same value as the listener)	modifiable manually to any available port	TCP	No

Table E–1   (Cont.) Ports Used in Oracle Components

Component and Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
Oracle Event Manager (EVM) Generates events for Oracle Clusterware. The port number is assigned automatically during installation. You cannot view or modify it afterward.	Dynamic	Dynamic	UDP	Yes
Oracle Grid Interprocess Communication (GIPCD) A support daemon that enables Redundant Interconnect Usage.	42424	Dynamic	UDP	Yes
Oracle Grid Naming Service (GNSD) The Oracle Grid Naming Service daemon provides a gateway between the cluster mDNS and external DNS servers. The gnsd process performs name resolution within the cluster.	53 (public), Dynamic (interconnect)	53 (public), Dynamic (interconnect)	UDP/TCP (public), TCP interconnect	No
Oracle HA Services daemon (OHASD) The Oracle High Availability Services (OHAS) daemon starts the Oracle Clusterware stack.	42424	Dynamic	UDP	Yes
Oracle Net Listener Allows Oracle client connections to the database by using Oracle Net Services. You can configure this port during installation. To reconfigure this port, use NETCA.	1521	Port number changes to the next available port. Modifiable manually to any available port.	TCP	No
Oracle Notification Services (ONS) Port for ONS, used to publish and subscribe service for communicating information about Fast Application Notification (FAN) events. The FAN notification process uses system events that Oracle Database publishes when cluster servers become unreachable or if network interfaces fail. Use srvctl to modify ONS ports.	6100 (local) 6200 (remote)	Configured manually	TCP	No

Table E–1   (Cont.) Ports Used in Oracle Components

Component and Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
Oracle Real Application Clusters The port number is assigned automatically during installation. You cannot view or modify it afterward.	Dynamic	Dynamic	UDP	Yes
Oracle XML DB - FTP The Oracle XML DB FTP is used when applications need to access an Oracle database from an FTP listener. The port is configured during installation and you *cannot* view it afterward. Refer to *Oracle XML DB Developer's Guide* for information about changing this port number.	0	Configured manually	FTP	No
Oracle XML DB- HTTP The Oracle XML DB HTTP port is used if Web-based applications need to access an Oracle database from an HTTP listener. The port is configured during installation and you *cannot* view it afterward. Refer to *Oracle XML DB Developer's Guide* for information about changing this port number.	0	Configured manually	HTTP	No

Index

Typical installation type,　2-4

U

undo management,　7-6
upgrade,　1-6
　of existing Oracle Databases,　3-2
　of listener,　3-2
　of Oracle Database with patch,　D-2
　Oracle Clusterware version requirement for,　1-6
　TIMESTAMP WITH TIME ZONE data,　1-16
upgrades
　and SCANs,　7-9
　and using same Oracle home,　1-7
　of existing Oracle databases,　2-3
　stopping processes,　1-19
user accounts
　postinstallation setup,　5-5
user authentication,　7-19
USERS tablespace
　described,　7-5

V

VIP addresses,　7-8

W

web browsers
　supported with Oracle Application Express,　1-12
　supported with Oracle Enterprise Manager,　1-12