Oracle® Real Application Clusters

Real Application Clusters Installation Guide

21c for Linux and UNIX

F31402-05

July 2025

ORACLE®

Oracle Real Application Clusters Real Application Clusters Installation Guide, 21c for Linux and UNIX

F31402-05

Primary Author: Subhash Chandra

Contributing Authors: Aparna Kamath, Janet Stern, Douglas Williams, Mark Bauer

Contributors: Jonathan Creighton, Anil Nair, Markus Michalewicz, Satish Panchumarthy, David Austin, Ian Cookson, 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, Michael Coulter, Rajesh Prasad, Kannan Viswanathan, Robert Achacoso, David Price, Ramesh Chakravarthula

Contents

4   Oracle Real Application Clusters Postinstallation Procedures

7 Removing Oracle Database Software

8 Using Scripts or Response Files to Create Oracle RAC Databases

9   Directory Structure for Oracle RAC Environments

10   Preparing to Upgrade an Existing Oracle RAC Database

11   Configuring Oracle Homes

12   Managing Oracle Database Port Numbers

Preface

This guide 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

- Documentation Accessibility

- Diversity and Inclusion

- Set Up Java Access Bridge to Implement Java Accessibility
  Install Java Access Bridge so that assistive technologies on Microsoft Windows systems can use the Java Accessibility API.

- Command Syntax
  Refer to these command syntax conventions to understand command examples in this guide.

- Conventions

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.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and

the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Set Up Java Access Bridge to Implement Java Accessibility

Install Java Access Bridge so that assistive technologies on Microsoft Windows systems can use the Java Accessibility API.

Java Access Bridge is a technology that enables Java applications and applets that implement the Java Accessibility API to be visible to assistive technologies on Microsoft Windows systems.

Refer to *Java Platform, Standard Edition Accessibility Guide* for information about the minimum supported versions of assistive technologies required to use Java Access Bridge. Also refer to this guide to obtain installation and testing instructions, and instructions for how to use Java Access Bridge.

Related Topics

- *Java Platform, Standard Edition Java Accessibility Guide*

Command Syntax

Refer to these command syntax conventions to understand command examples in this guide.

Convention	Description
$	Bourne or BASH shell prompt in a command example. Do not enter the prompt as part of the command.
%	C Shell prompt in a command example. Do not enter the prompt as part of the command.
#	Superuser (root) prompt in a command example. Do not enter the prompt as part of the command.
monospace	UNIX command syntax
backslash \	A backslash is the UNIX and Linux command continuation character. It is used in command examples that are too long to fit on a single line. Enter the command as displayed (with a backslash) or enter it on a single line without a backslash: dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b \ count=10000
braces { }	Braces indicate required items: .DEFINE {macro1}
brackets [ ]	Brackets indicate optional items: cvtcrt termname [outfile]
ellipses ...	Ellipses indicate an arbitrary number of similar items: CHKVAL fieldname value1 value2 ... valueN

Convention	Description
italic	Italic type indicates a variable. Substitute a value for the variable: *library_name*
vertical line |	A vertical line indicates a choice within braces or brackets: FILE filesize [K|M]

Conventions

The following text conventions are used in this document:

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

1
Oracle RAC Installation Checklist

Review these checklists for installing Oracle Real Application Clusters (Oracle RAC).

- **Deployment Checklist for Oracle RAC Database**
  Use this checklist to decide the deployment method for Oracle Database.

- **Server Hardware and Software Review Checklist for Oracle RAC Installation**
  Review this checklist to ensure you have met the minimum hardware and software requirements for Oracle RAC.

- **Installer Planning Checklist for Oracle Database**
  Use this checklist to assist you to be prepared before starting Oracle Universal Installer.

- **Oracle RAC Upgrade Checklist**
  Review this checklist to prepare your Oracle Real Application Clusters (Oracle RAC) databases for upgrade.

Deployment Checklist for Oracle RAC Database

Use this checklist to decide the deployment method for Oracle Database.

Table 1-1   Deployment Checklist for Oracle RAC Database

Check	Task
Deploy Oracle RAC or Oracle RAC One Node software	Use one of the following deployment methods: • Install Oracle RAC or Oracle RAC One Node Database software using Oracle Universal Installer (OUI). • Provision Oracle RAC Or Oracle RAC One Node Database software using Oracle Fleet Patching and Provisioning.
Deploy Oracle Database software and create Oracle RAC or Oracle RAC One Node databases	Use one of the following deployment methods: • Install Oracle RAC or Oracle RAC One Node Database using Oracle Universal Installer (OUI). • Provision Oracle RAC Or Oracle RAC One Node Database using Oracle Fleet Patching and Provisioning.
Create Oracle RAC or Oracle RAC One Node Database in an already-installed Oracle home	• Use Database Configuration Assistant (DBCA). • Use Oracle Fleet Patching and Provisioning.

Related Topics

- **Installing Oracle RAC and Oracle RAC One Node**
  After installing Oracle Clusterware, as described in *Oracle Grid Infrastructure Installation Guide* for your platform, you can install Oracle RAC Database software.

- **About Deploying Oracle Databases Using Oracle Fleet Patching and Provisioning**
  Provision, patch, and upgrade software homes on any number of nodes from a single cluster with standard gold images using Oracle FPP.

Server Hardware and Software Review Checklist for Oracle RAC Installation

Review this checklist to ensure you have met the minimum hardware and software requirements for Oracle RAC.

Table 1-2   Hardware and Software Review Checklist for Oracle RAC

Check	Task
Operating System General Requirements	Prepare servers as described in *Oracle Grid Infrastructure Installation Guide* for your platform.
Kernel Requirements and Fixup Scripts	During an installation, Oracle Universal Installer (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.
Review Existing Oracle Installations and Upgrade Plans	To install Oracle RAC 21c, you must install Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) 21c on your cluster. The Oracle Clusterware version must be equal to or greater than the Oracle RAC version that you plan to install.
	For the most current updates and best practices about pre-upgrade, post-upgrade, compatibility, and interoperability discussions, refer to Support Note 1670757.1 on My Oracle Support.
	See Also:
	Oracle Database Upgrade Guide
	Oracle Grid Infrastructure Installation Guide
Server Hardware on each node	Use identical server hardware on each node, to simplify server maintenance.
Time Zone Requirement	Upgrade the Time Zone File and TSTZ Data:
	As part of an installation of Oracle Database 21c, time zone version files from 1 to 12 are installed in the following 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.
	See *Oracle Database Globalization Support Guide* for information about how to upgrade the time zone file and TSTZ data
Use Cluster Verification Utility (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. Contact your system or storage administrators to have them fix errors manually. CVU is available in the Grid home, in the bin directory.

Table 1-2   (Cont.) Hardware and Software Review Checklist for Oracle RAC

Check	Task
Ensure that Oracle RAC Databases You are Installing are Compatible with Existing Databases	• 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 21c when you perform an Oracle Grid Infrastructure 21c installation. • If you have an existing Oracle home, then you can create a new Oracle home and install Oracle Database 21c into the new Oracle home. 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 home directory for Oracle Database.
Platform-Specific Server Configuration	• Ensure an external jobs user is created for HP-UX.

Related Topics

• *Oracle Database Globalization Support Guide*

• *Oracle Database Upgrade Guide*

• *Oracle Grid Infrastructure Installation and Upgrade Guide*

Installer Planning Checklist for Oracle Database

Use this checklist to assist you to be prepared before starting Oracle Universal Installer.

Table 1-3   Oracle Universal Installer Planning Checklist for Oracle Database Installation

Check	Task
Read the Release Notes	Review release notes for your platform, which are available for your release at the following URL: http://docs.oracle.com/en/database/database.html
Review Oracle Support Certification Matrix	New platforms and operating system software versions may be certified after this guide is published, review the certification matrix on the My Oracle Support website for the most up-to-date list of certified hardware platforms and operating system versions: https://support.oracle.com/ You must register online before using My Oracle Support. After logging in, from the menu options, select the **Certifications** tab. On the Certifications page, use the **Certification Search** options to search by Product, Release, and Platform. You can also search using the **Certification Quick Link** options such as Product Delivery, and Lifetime Support.
Review the Licensing Information	You are permitted to use only those components in Oracle Database for which you have purchased licenses. For more information about licenses, refer to the following URL: *Oracle Database Licensing Information*

Table 1-3   (Cont.) Oracle Universal Installer Planning Checklist for Oracle Database Installation

Check	Task
Run OUI with CVU and use fixup scripts	Oracle Universal Installer is fully integrated with Cluster Verification Utility (CVU), automating many CVU prerequisite checks. Oracle Universal Installer runs all prerequisite checks and creates fixup scripts when you run the installer. You can run OUI up to the Summary screen without starting the installation.
	You can also run CVU commands manually to check system readiness. For more information, see:
	Oracle Clusterware Administration and Deployment Guide
Update and run Oracle ORAchk for runtime and upgrade checks, or runtime health checks	Oracle recommends that you update to the latest version of Oracle ORAchk by installing the latest version of Autonomous Health Framework (AHF).
	The Oracle ORAchk utility provides system checks that can help to prevent issues before and after installation. These checks include kernel requirements, operating system resource allocations, and other system requirements.
	Use the Oracle ORAchk Upgrade Readiness Assessment to obtain an automated upgrade-specific system health check for upgrades to 21c. For example:
	• Before you perform a fresh database installation:
	./orachk -profile preinstall
	• To upgrade your existing database to a higher version or release:
	./orachk -pregupgrade -targetversion 21.0.0.0.0
	The Oracle ORAchk Upgrade Readiness Assessment automates many of the manual pre- and post-upgrade checks described in Oracle upgrade documentation. Check My Oracle Support Note 2550798.1 for more information about Oracle ORAchk support.
	https://support.oracle.com/epmos/faces/DocContentDisplay?id=2550798.1&parent=DOCUMENTATION&sourceId=USERGUIDE
Verify if Oracle Grid Infrastructure is installed	For Oracle Real Application Clusters (Oracle RAC) installations, ensure that you have installed and configured Oracle Grid Infrastructure for a cluster.
	If you want to use Oracle ASM or Oracle Restart, then install Oracle Grid Infrastructure for a standalone server before you install and create the database. Otherwise, to use Oracle ASM, you must complete an Oracle Grid Infrastructure installation, and then manually register the database with Oracle Restart.
Check running Oracle processes, and shut down if necessary	• On a standalone database not using Oracle ASM: You do not need to shut down the database while you install Oracle Grid Infrastructure.
	• On a standalone database using Oracle ASM: The Oracle ASM instances are restarted during installation.
	• On an Oracle RAC Database node: This installation requires an upgrade of Oracle Clusterware, as Oracle Clusterware is required to run Oracle RAC. As part of the upgrade, you must shut down the database one node at a time as the rolling upgrade proceeds from node to node.

Table 1-3   (Cont.) Oracle Universal Installer Planning Checklist for Oracle Database Installation

Check	Task
Ensure cron jobs do not run during installation	If the installer is running when daily cron jobs start, then you may encounter unexplained installation problems if your cron job is performing cleanup, and temporary files are deleted before the installation is finished. Oracle recommends that you complete installation before daily cron jobs are run, or disable daily cron jobs that perform cleanup until after the installation is completed.
Obtain your My Oracle Support account information.	During installation, you may require a My Oracle Support user name and password to configure updates, download software updates, and other installation tasks. You can register for My Oracle Support at the following URL: https://support.oracle.com/
Decide Oracle Database management tool	By default, Oracle Database is managed by Oracle Enterprise Manager Database Express. If you have an existing Oracle Management Agent, and decide to use Oracle Enterprise Manager Cloud Control to centrally manage your database, then obtain the following information to enter during the database installation: • OMS host • OMS port • EM admin username • EM admin password • Specify password of ASMSNMP user For more information, see: • *Oracle Enterprise Manager Cloud Control Administrator's Guide*
Review memory allocation and Automatic Memory Management feature	You can enable automatic memory management either during, or after Oracle Database installation. If you enable automatic memory management after installation, then you must shut down and restart the database. If the total physical memory of your database instance is greater than 4 GB, then you cannot select the Oracle Automatic Memory Management option during database installation and creation. Instead, use automatic shared memory management. Automatic shared memory management automatically distributes the available memory among the various components as required, allowing the system to maximize the use of all available SGA memory. For more information, see: *Oracle Database Administrator's Guide*

Table 1-3   (Cont.) Oracle Universal Installer Planning Checklist for Oracle Database Installation

Check	Task
Determine superuser (root) privilege delegation option for installation	During a database or grid infrastructure installation, you are asked to run configuration scripts as the root user.
	You can either run these scripts manually as root when prompted, or you can provide configuration information and passwords using a root privilege delegation option.
	To run root scripts automatically, select **Automatically run configuration scripts** during installation.
	• **Use root user credentials** Provide the superuser password for cluster member node servers.
	• **Use Sudo** Sudo is a UNIX and Linux utility that allows members of the sudoers list privileges to run individual commands as root. Provide the username and password of an operating system user that is a member of sudoers, and is authorized to run Sudo on each cluster member node.
	To enable Sudo, have a system administrator with the appropriate privileges configure a user that is a member of the sudoers list, and provide the username and password when prompted during installation.
Oracle Database Client and Oracle Database interoperability	For information about interoperability between Oracle Database Client and Oracle Database releases, see My Oracle Support Note 207303.1: https://support.oracle.com/rs?type=doc&id=207303.1

Oracle RAC Upgrade Checklist

Review this checklist to prepare your Oracle Real Application Clusters (Oracle RAC) databases for upgrade.

Table 1-4   Oracle RAC Upgrade Checklist

Check	Task
Review Existing Oracle Installations and Upgrade Plans	To install Oracle RAC 21c, you must install Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) 21c 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 Database Upgrade Guide
	Oracle Grid Infrastructure Installation Guide

Table 1-4   (Cont.) Oracle RAC Upgrade Checklist

Check	Task
Ensure that Oracle RAC Databases You are Installing are Compatible with Existing Databases	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 21c when you perform an Oracle Grid Infrastructure 21c installation.
	If you have an existing Oracle home, then you can create a new Oracle home and install Oracle Database 21c into the new Oracle home. Ensure that Oracle Clusterware is in a separate Oracle Grid Infrastructure home. Oracle Grid Infrastructure for cluster installations cannot be installed in the Oracle home directory for Oracle Database.
Migrate files from RAW devices to Supported Storage	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.
Prepare to upgrade all existing nodes	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 upgrade all three nodes. You cannot upgrade only two nodes of the cluster, removing the third instance in the upgrade.
Ensure that Oracle RAC database version is equal to or older than the version of Oracle Clusterware	Before you upgrade Oracle RAC database, ensure that Oracle Grid Infrastructure is upgraded to an equal or a later release.
	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. For example: • If your servers use Oracle Grid Infrastructure 21c, then you can have an Oracle Database 21c single-instance database running on one node, and separate Oracle RAC 18c, Oracle RAC 19c, or Oracle RAC 21c databases also running on the cluster. • You cannot have Oracle Grid Infrastructure 19c installed on your cluster, and install Oracle RAC 21c.

2

Installing Oracle RAC and Oracle RAC One Node

After installing Oracle Clusterware, as described in *Oracle Grid Infrastructure Installation Guide* for your platform, you can install Oracle RAC Database software.

- [About Image-Based Oracle Database Installation](#)
  Understand image-based installation to simplify installation and configuration of Oracle Database software.

- [Setup Wizard Installation Options for Creating Images](#)
  Gold image-creation options to use with your Oracle Database or Oracle Grid Infrastructure installation setup wizards.

- [Downloading the Software from Oracle Software Delivery Cloud Portal](#)
  You can download the software from Oracle Software Delivery Cloud.

- [Installing Oracle RAC and Oracle RAC One Node Databases](#)
  Understand the process to install Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node Databases.

- [About Deploying Oracle Databases Using Oracle Fleet Patching and Provisioning](#)
  Provision, patch, and upgrade software homes on any number of nodes from a single cluster with standard gold images using Oracle FPP.

About Image-Based Oracle Database Installation

Understand image-based installation to simplify installation and configuration of Oracle Database software.

To install Oracle Database, create the new Oracle home, extract the image file into the newly-created Oracle home, and run the setup wizard to register the Oracle Database product.

Using image-based installation, you can install and upgrade Oracle Database for single-instance and cluster configurations. If you install or clone an Oracle Database image, then all Oracle Database options such as Oracle OLAP ($olap$) and Oracle Real Application Testing (rat) are enabled by default.

This installation feature streamlines the installation process and supports automation of large-scale custom deployments. You can also use this installation method for deployment of customized images, after you patch the base-release software with the necessary Release Updates (Updates) or Monthly Recommended Patches (MRPs).

> ⓘ **Note**
>
> You must extract the image software ($db\_home.zip$) into the directory where you want your Oracle Database home to be located, and then run the Oracle Database Setup Wizard to start the Oracle Database installation and configuration. Oracle recommends that the Oracle home directory path you create is in compliance with the Oracle Optimal Flexible Architecture recommendations.

Setup Wizard Installation Options for Creating Images

Gold image-creation options to use with your Oracle Database or Oracle Grid Infrastructure installation setup wizards.

Options

In image-based installations, you can start your Oracle Database installation or Oracle Grid Infrastructure installations by running the setup wizards runInstaller and gridSetup.sh respectively. Both these wizards come with the following image-creation options.

Table 2-1   Image-Creation Options for Setup Wizard

Option	Description
-createGoldImage	Creates a gold image from the current Oracle home.
-destinationLocation	Specify the complete path, or location, where the gold image will be created.
-exclFiles	Specify the complete paths to the files to be excluded from the newly created gold image.
—help	Displays help for all the available options.

For example:

```
cd $ORACLE_HOME
./runInstaller -createGoldImage -destinationLocation /tmp/my_db_images -exclFiles /u01/app/oracle/product/21.0.0/dbhome_1/relnotes
```

```
cd Grid_home
./gridSetup.sh -createGoldImage -destinationLocation /tmp/my_grid_images -exclFiles /u01/app/oracle/product/21.0.0/dbhome_1/relnotes
```

Where:

/tmp/my_db_images is a temporary file location where the image zip file is created.

/tmp/my_grid_images is a temporary file location where the image zip file is created.

/u01/app/oracle/product/21.0.0/dbhome_1/relnotes is the file to be excluded from the newly created gold image.

Downloading the Software from Oracle Software Delivery Cloud Portal

You can download the software from Oracle Software Delivery Cloud.

1. Use a browser to access the Oracle Software Delivery Cloud portal:

   https://edelivery.oracle.com/

2. Click **Sign In** and enter your Oracle account username and password.

3. Type **Oracle Database** in the search bar. Click the **Add to Cart** button corresponding to the Oracle Database version that you want to download

4. In the Checkout page, click **Checkout** and deselect any products that you do not want to download.

5. Select the operating system platform on which you want to install the software from the **Platform/Languages** column.

6. Click **Continue**.

7. Review the license agreement.

8. Select the **I reviewed and accept the Oracle License Agreement** checkbox. Click **Continue**.

9. Click **Download** to start downloading the software.

10. After you download the files, click **View Digest** to verify that the checksum matches the value listed on the download page.

Installing Oracle RAC and Oracle RAC One Node Databases

Understand the process to install Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node Databases.

- [Steps to Install Oracle Database](#)
  Installation of Oracle Oracle RAC and Oracle RAC One Node Databases requires software-only installation followed by database creation using Oracle DBCA.

- [Installing Oracle RAC and Oracle RAC One Node Database Software](#)
  Install Oracle RAC or Oracle RAC One Node software using the gold image-based installation with Oracle Database setup wizard.

Steps to Install Oracle Database

Installation of Oracle Oracle RAC and Oracle RAC One Node Databases requires software-only installation followed by database creation using Oracle DBCA.

Installation of Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node Databases is a two-step process:

1. Install Oracle RAC software by running the Oracle Database installer.

2. Create and Configure Oracle RAC and Oracle RAC One Node Databases using Database Configuration Assistant (DBCA).

Installing Oracle RAC and Oracle RAC One Node Database Software

Install Oracle RAC or Oracle RAC One Node software using the gold image-based installation with Oracle Database setup wizard.

Oracle Real Application Clusters Database installation is a two-step process. This procedure describes the first step — to install Oracle RAC software.

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.

Ensure that you have su or sudo credentials, because you will be required to either provide root credentials to automatically run the root scripts or run a script as the root user during installation.

1. Ensure that you can access other nodes with SSH. The installer requires that the user account running the Oracle RAC installation is permitted to set up passwordless SSH. The installer 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 the installer, 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 prevent automatic SSH configuration from within the installer, then you must ensure that SSH is configured and enabled before you proceed to start installation.

3. Download the Oracle Database installation image files (db_home.zip) and extract the files into a new Oracle home directory. For example:

```
$ mkdir -p /u01/app/oracle/product/21.0.0/dbhome_1
$ chgrp oinstall /u01/app/oracle/product/21.0.0/dbhome_1
$ cd /u01/app/oracle/product/21.0.0/dbhome_1
$ unzip -q /tmp/db_home.zip
```

> ⓘ **Note**
>
> • Oracle recommends that the Oracle home directory path you create is in compliance with the Oracle Optimal Flexible Architecture recommendations. Also, unzip the installation image files only in this Oracle home directory that you created.
>
> • Oracle home or Oracle base cannot be symlinks, nor can any of their parent directories, all the way to up to the root directory.

4. From the Oracle home directory, start the Oracle Database software installation:

```
$ cd /u01/app/oracle/product/21.0.0/dbhome_1
$ ./runInstaller
```

> ⓘ **Note**
>
> Run the runInstaller command from the Oracle home directory only. Do not run runInstaller command that resides at $ORACLE_HOME/oui/bin/, or any other location, to install Oracle Real Application Clusters or Oracle Grid Infrastructure.

5. In the Select Configuration Option screen, select the **Setup Software Only** option. Click **Next**.

6. In the Select Database Installation Option screen, select the **Oracle Real Application Clusters Database Installation** option. Click **Next**.

   Starting with Oracle Database 21c, installation of non-CDB Oracle Database architecture is no longer supported.

The non-CDB architecture was deprecated in Oracle Database 12c. It is desupported in Oracle Database 21c. Oracle Universal Installer can no longer be used to create non-CDB Oracle Database instances.

> ⓘ **Note**
>
> A multitenant container database is the only supported architecture in Oracle Database 21c. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

7. In the Node Selection screen, select all the nodes where you want to install Oracle RAC or Oracle RAC One Node software. Click **Next**.

8. In the Select Database Edition screen, select the **Enterprise Edition** option. Click **Next**.

9. In the Specify Install Location screen, provide the location of Oracle base for Oracle RAC software installation.

   The Oracle base directory must be different from the Grid home directory. Click **Next**.

10. In the Privileged Operating System Groups screen, accept the default operating system group names for Oracle Database administration and click **Next**.

11. Select the option to **Automatically run configuration scripts**. Enter the credentials for the root user or a sudo account, then click **Next**.

    Alternatively, you can run the scripts manually as the root user at the end of the installation process when prompted by the installer.

12. The Perform Prerequisite Checks screen displays the results of the prerequisites checks. If any of the checks have a status of Failed and are not Fixable, then you must manually correct these issues. After you have fixed the issue, you can click the Check Again button to have the installer recheck the requirement and update the status. Repeat as needed until all the checks have a status of Succeeded. Click **Next**.

13. Review the contents of the Summary screen and then click **Install**.

    The installer displays a progress indicator enabling you to monitor the installation process.

14. If you did not configure automation of the root scripts, then you are required to run root.sh script as the root user, as specified in the Execute Configuration Scripts window. Do *not* click **OK** until you have run the root.sh script on all nodes as directed.

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

After installing Oracle RAC software, run Database Configuration Assistant (DBCA) from the ORACLE_HOME/bin/dbca directory to create and configure Oracle RAC databases.

Related Topics

- [About Deploying Oracle Databases Using Oracle Fleet Patching and Provisioning](#)
  Provision, patch, and upgrade software homes on any number of nodes from a single cluster with standard gold images using Oracle FPP.

- [Oracle Real Application Clusters Postinstallation Procedures](#)
  Review this information to complete the postinstallation tasks after you have installed Oracle Real Application Clusters (Oracle RAC).

Related Topics

*   [Creating Oracle RAC or Oracle RAC One Node Databases with Oracle DBCA](#)
    Use Oracle Database Configuration Assistant (DBCA) in standalone mode to create and delete Oracle Real Application Clusters (Oracle RAC) databases.

About Deploying Oracle Databases Using Oracle Fleet Patching and Provisioning

Provision, patch, and upgrade software homes on any number of nodes from a single cluster with standard gold images using Oracle FPP.

> ⓘ **Note**
>
> Starting with Oracle Database 19c, Rapid Home Provisioning feature is known as Oracle Fleet Patching and Provisioning or Oracle FPP.

With Oracle Fleet Patching and Provisioning (Oracle FPP), you create, store, and manage templates of Oracle homes as images (called gold images) of Oracle Databases, Grid Infrastructure, middleware, and applications. These gold images are maintained on an Oracle Fleet Patching and Provisioning Server, which can provision and maintain working copies of the images on any number of nodes in an Information Technology estate.

Oracle Fleet Patching and Provisioning

Deploying Oracle software using Oracle FPP has the following advantages:

*   Enables you to create and manage Oracle Real Application Clusters (Oracle RAC), single instance, and Oracle Real Application Clusters One Node (Oracle RAC One Node) databases. You can deploy single-instance databases with or without Oracle Grid Infrastructure.

*   Ensures standardization and enables high degrees of automation with gold images and managed lineage of deployed software.

*   Supports change management. With standardized Oracle homes, an administrator has better control of the hosted Oracle software and can easily manage the mass deployment, patching, and upgrade of the software through a single location for change management.

*   Minimizes downtime during patching and upgrades, eases rollbacks, and makes provisioning for large systems easier and more efficient.

*   Simplifies maintenance and patching of database software.

*   Reduces the cumulative time to patch software images, since a single Oracle home may be used for many database instances.

> ⓘ **See Also**
>
> *Oracle Clusterware Administration and Deployment Guide* for information about setting up the Oracle Fleet Patching and Provisioning Server and Client, creating and using gold images for provisioning and patching Oracle Database homes.

3

Creating Oracle RAC or Oracle RAC One Node Databases with Oracle DBCA

Use Oracle Database Configuration Assistant (DBCA) in standalone mode to create and delete Oracle Real Application Clusters (Oracle RAC) databases.

- **About Oracle Database Configuration Assistant**
  Understand the features Oracle Database Configuration Assistant (Oracle DBCA) offers for creating your Oracle RAC databases.

- **Selecting Installation Options for Oracle RAC**
  Review the topics to select options for installing Oracle RAC.

- **Automatic Listener Migration from Earlier Releases**
  Review this information for listener migration from earlier database releases.

- **Verifying Requirements for Oracle DBCA**
  Use CVU to verify that your system is prepared for configuration changes before using Oracle DBCA to create Oracle RAC databases.

- **Tasks to Complete Before Using Oracle DBCA to Create any Oracle RAC Database**
  Configure your system to meet the software requirements, if you did not complete your system configuration during the Oracle Grid Infrastructure installation.

- **Selecting Oracle DBCA Options to Create an Oracle RAC or Oracle RAC One Node Database**
  Review this information to use Oracle Database Configuration Assistant (Oracle DBCA) to create an Oracle RAC or Oracle RAC One Node Database.

- **Using DBCA to Create an Oracle RAC One Node Database**
  Configure Oracle RAC One Node database using Oracle DBCA after creating a software-only setup of Oracle RAC software on cluster nodes.

- **Installing the Oracle Database Vault Option**
  Installing and configuring Oracle Database Vault requires actions during and after installation.

- **Deleting an Oracle RAC Database Using Oracle DBCA**
  Delete an Oracle RAC database using Oracle Database Configuration Assistant (Oracle DBCA) with all of the database components.

About Oracle Database Configuration Assistant

Understand the features Oracle Database Configuration Assistant (Oracle DBCA) offers for creating your Oracle RAC databases.

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

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

You can use Oracle 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 supports. If you are not sure which to choose, then select the default **General Purpose or Transaction Processing** template.

Selecting Installation Options for Oracle RAC

Review the topics to select options for installing Oracle RAC.

- [About Cluster Node Selection for Database Installation](#)
  Select the cluster nodes to install Oracle RAC or Oracle RAC One Node databases.

- [Selecting a Database Name](#)
  The database name is comprised of various strings and must contain only permitted characters. Review the following guidelines when selecting a database name.

- [Requirements for Database Passwords](#)
  To secure your database, use passwords that satisfy the Oracle recommended password requirements, even the passwords for predefined user accounts.

- [About Automatic Memory Management Installation Options](#)
  Decide if you want to configure Automatic Memory Management during installation.

- [About Character Set Selection During Installation](#)
  Before you create the database, decide the character set that you want to use.

- [Managing Oracle RAC Database Services After Installation](#)
  Administer and monitor database services using SRVCTL, Oracle Enterprise Manager Database Express, or Oracle Enterprise Manager Cloud Control.

About Cluster Node Selection for Database Installation

Select the cluster nodes to install Oracle RAC or Oracle RAC One Node databases.

You can use cluster nodes to host Oracle RAC database instances that run in read-only mode, which become Oracle RAC Reader Nodes. Oracle RAC Reader Nodes facilitate Oracle Flex Cluster architecture by allocating a set of read/write instances running Online Transaction Processing (OLTP) workloads and a set of read-only database instances across cluster nodes.

Selecting a Database Name

The database name is comprised of various strings and must contain only permitted characters. Review the following guidelines when 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.

> ⓘ **Note**
>
> For Oracle Real Applications Cluster (Oracle RAC) databases, the pluggable database (PDB) name must be unique in the cluster.

Database Name and ORACLE_SID

The Oracle Service Identifier (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 prefix can contain up to 12 characters. 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 administator-managed database, then Oracle Database Configuration Assistant generates the default SID for the instance names, using the format *name#*, where name is the first eight alphanumeric characters of DB_UNIQUE_NAME, and # is the instance number. However, during installation or database creation you can specify a nondefault value for the SID. The instance number is automatically added to the end of this string for each instance.

For an Oracle RAC One Node database, the instance name 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.

Example 3-1   Global Database Name and Related Initialization Parameters

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:

Parameter	Value
DB_UNIQUE_NAME	orl$racprod2551
DB_DOMAIN	example.com
DB_NAME	orl$racp

Example 3-2   DB_UNIQUE_NAME and Related ORACLE_SID Values

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
Administrator-managed Oracle RAC instance	orlracpr1
Oracle RAC One Node database instance	orlracpr_1

Requirements for Database Passwords

To secure your database, use passwords that satisfy the Oracle recommended password requirements, even the passwords for predefined user accounts.

Oracle Database provides a set of predefined user accounts. Create passwords in a secure fashion. If you have default passwords, change these passwords to secure passwords.

You can manage the security for Oracle Database users in various ways:

• Enforce restrictions on the way that passwords are created
• Create user profiles
• Use user resource limits to further secure user accounts

Related Topics

• *Oracle Database Security Guide*

About Automatic Memory Management Installation Options

Decide if you want to configure Automatic Memory Management during installation.

During a **Typical** installation, you create your database with Oracle 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.

If the total physical memory of your database instance is greater than 4 GB, then you cannot select the Oracle Automatic Memory Management option during database installation and creation. Instead, use automatic shared memory management. Automatic shared memory management automatically distributes the available memory among the various components as required, allowing the system to maximize the use of all available SGA memory.

With automatic memory management, the Oracle Database 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.

You can enable automatic memory management either during, or after the database installation. Enabling automatic memory management after installation involves a shutdown and restart of the database.

> ⓘ **Note**
>
> By default, automatic memory management is disabled when you perform typical installation on a node that has more than 4 GB of RAM.

Related Topics

• *Oracle Database Administrator's Guide*

About Character Set Selection During Installation

Before you create the database, decide the character set that you want to use.

After a database is created, changing its character set is usually very expensive in terms of time and resources. Such operations may require converting all character data by exporting the whole database and importing it back. Therefore, it is important that you carefully select the database character set at installation time.

Oracle Database uses character sets for the following:

• Data stored in SQL character data types (CHAR, VARCHAR2, CLOB, and LONG).

• Identifiers such as table names, column names, and PL/SQL variables.

• Stored SQL and PL/SQL source code, including text literals embedded in this code.

The default database character set of a database created from the General Purpose/ Transaction Processing or the Data Warehousing template is Unicode AL32UTF8.

Unicode is the universal character set that supports most of the currently spoken languages of the world. It 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 supporting the Internet and the global economy.

Because AL32UTF8 is a multibyte character set, it requires slightly more CPU time for text processing compared to single-byte character sets. Also, storage space requirements are higher for text in most languages compared to corresponding legacy character sets. However, the universality and flexibility of Unicode that enables addition of data in new languages to applications running in an AL32UTF8 database generally outweighs these additional costs.

The database character set of an Oracle Database, that is, of its CDB$ROOT container, determines which pluggable databases (PDBs) can be plugged into it. If you use Unicode AL32UTF8 as your database character set, then you can plug in a PDB in any database character set supported by Oracle Database (with the exception of EBCDIC-based character sets). If you use any character set other than AL32UTF8 when creating the container database, you will be able to plug in PDBs in the same character set only. Therefore, you should generally use the default option for the database character set when installing a new database.

If you need to deploy PDBs in a given legacy character set to fulfill a specific compatibility, storage, or performance requirement, create a temporary container database in this legacy character set with one empty PDB. This PDB will have the same legacy database character set. Then, unplug this PDB and plug it into the target AL32UTF8 container database. Drop the temporary container database. You can use such a plugged-in PDB as a template to clone further PDBs in the same legacy character set as needed. You can use the same method to add further legacy character set template PDBs to the same AL32UTF8 container database, as required.

> ⓘ **See Also**
>
> *Oracle Database Globalization Support Guide* for more information about choosing a database character set for an Oracle Database.

Managing Oracle RAC Database Services After Installation

Administer and monitor database services using SRVCTL, Oracle Enterprise Manager Database Express, or Oracle Enterprise Manager Cloud Control.

You cannot use Oracle Database Configuration Assistant (DBCA) to manage database services for Oracle Real Application Clusters (Oracle RAC) databases.

The Server Control (SRVCTL) utility is installed on each node by default. Oracle Enterprise Manager Cloud Control is available separately with the Oracle Enterprise Manager Cloud Control installation, and on the Oracle Technology Network website at:

https://www.oracle.com/downloads/

> ⓘ **See Also**
>
> *Oracle Enterprise Manager Online Help* for service management using Oracle Enterprise Manager

Automatic Listener Migration from Earlier Releases

Review this information for listener migration from earlier database releases.

If your system has an Oracle Grid Infrastructure 12c Release 2 (12.2), 18c, or 19c installation, and you install Oracle Grid Infrastructure 21c either to coexist with or to upgrade the Oracle Grid Infrastructure 12.2, 18c, or 19c installation, then most installation types automatically migrate the existing listener to the 21c 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 ($LISTENER_{nodename}$) is migrated to the Oracle Grid Infrastructure home (Grid home). Oracle Database Configuration Assistant always uses the default listener.

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

Verifying Requirements for Oracle DBCA

Use CVU to verify that your system is prepared for configuration changes before using Oracle DBCA to create Oracle RAC databases.

1. To use Oracle Database Configuration Assistant (Oracle 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:

   /u01/app/21.0.0/grid/bin/cluvfy stage -pre dbcfg -fixup -n *node_list* -d *ORACLE_HOME* [-verbose]

   In the preceding syntax example, 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.

2. To verify that your system is prepared for an Oracle Database with Oracle Real Application Clusters (Oracle RAC) installation on a two-node cluster with nodes *node1* and *node2*, with the Oracle Grid Infrastructure home path/u01/app/21.0.0/grid, and with the Oracle home path /u01/app/oracle/product/21.0.0/dbhome_1, enter the following command:

   $ /u01/app/21.0.0/grid/bin/cluvfy stage -pre dbcfg -fixup -n node1,node2 -d /u01/app/oracle/product/21.0.0/dbhome_1

   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 CVU summary 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.

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

Configure your system to meet the software requirements, if you did not complete your system configuration during the Oracle Grid Infrastructure installation.

- [Load SSH Keys Into Memory Before Starting DBCA](#)
  Load SSH keys into memory for the terminal session where you start Oracle DBCA, to avoid user equivalency errors while starting Oracle DBCA.

- [Decide on a Naming Convention to Use for Your Oracle RAC Database](#)
  Review this information for naming conventions for Oracle RAC database.

- [Configure Shared Storage for the Oracle RAC Database](#)
  Install Oracle Grid Infrastructure for a cluster and configure shared storage for Oracle RAC files before creating an Oracle RAC database using Oracle DBCA.

Load SSH Keys Into Memory Before Starting DBCA

Load SSH keys into memory for the terminal session where you start Oracle DBCA, to avoid user equivalency errors while starting Oracle 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 Oracle Database Configuration Assistant (Oracle DBCA).

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

Review this information for naming conventions for 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.

Configure Shared Storage for the Oracle RAC Database

Install Oracle Grid Infrastructure for a cluster and configure shared storage for Oracle RAC files before creating an Oracle RAC database using Oracle DBCA.

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 and Upgrade Guide for Linux* for your platform for more information about shared storage configuration requirements.

Selecting Oracle DBCA Options to Create an Oracle RAC or Oracle RAC One Node Database

Review this information to use Oracle Database Configuration Assistant (Oracle DBCA) to create an Oracle RAC or Oracle RAC One Node Database.

> ⓘ **Note**
>
> You can no longer set up email notification for Oracle Real Application Clusters (Oracle RAC) databases either from Oracle DBCA or Oracle Universal Installer (OUI).

- Starting Oracle DBCA
  Start Oracle Database Configuration Assistant (Oracle DBCA) in GUI mode to create and manage Oracle RAC databases.

- Cluster Detection and Node Selection when Using Oracle DBCA
  Oracle DBCA automatically shows Oracle RAC options at startup if it detects from the central Oracle Inventory that the Oracle home is enabled for Oracle RAC.

- [Using Oracle DBCA to Select Storage to Use With any Oracle RAC Database](#)
  Choose from Oracle ASM disk groups or a supported cluster file system to store Oracle RAC database files.

- [Selecting Server Pool Option with Oracle DBCA](#)
  Choose existing server pools to host policy-managed databases while creating a database using Oracle Database Configuration Assistant (Oracle DBCA).

- [Using DBCA to Specify Database Initialization Parameters for Oracle RAC](#)
  Specify initialization parameters if you intend to add more nodes in your cluster.

- [Actions Performed By DBCA for Oracle RAC Databases](#)
  Review this information to understand about DBCA actions during Oracle RAC database creation.

Starting Oracle DBCA

Start Oracle Database Configuration Assistant (Oracle DBCA) in GUI mode to create and manage Oracle RAC databases.

1. Log in as the software installation owner user, for example oracle.

2. Start Oracle DBCA by running the following command:

   $ $ORACLE_HOME/bin/dbca

> ⓘ **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.

Cluster Detection and Node Selection when Using Oracle DBCA

Oracle DBCA automatically shows Oracle RAC options at startup 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, perform the following steps to make the nodes visible to Oracle DBCA.

1. Check that the OUI inventory is correctly located in the directory /etc/oraInst.loc, and that the oraInventory file is not corrupted.

2. Perform Oracle Clusterware diagnostics by using the following CVU command:

   $ /u01/app/21.0.0/grid/bin/cluvfy/cluvfy.sh stage -post crsinst -n *nodelist*

   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).

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

Choose from Oracle ASM disk groups or a supported cluster file system to store Oracle RAC database files.

On the **Specify Database Storage Options** page, if you do not see the disk groups in Oracle Database Configuration Assistant (Oracle DBCA), then either Oracle ASM is not configured, or disk groups are not mounted. You can create disk groups using Automatic Storage Management Configuration Assistant (ASMCA) in the Oracle Grid Infrastructure home before starting Oracle 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 (FRA) defaults to the Oracle ASM disk group.

Selecting Server Pool Option with Oracle DBCA

Choose existing server pools to host policy-managed databases while creating a database using Oracle Database Configuration Assistant (Oracle DBCA).

During Database creation, you have the option to select an existing server pool or create a new server pool.

Starting with Oracle Grid Infrastructure 21c, policy-managed databases are deprecated.

You can continue to use existing server pools, and create new pools and policies. Resources using existing server pools can continue to use them transparently.

Using DBCA to Specify Database Initialization Parameters for Oracle RAC

Specify initialization parameters if you intend to add more nodes in your cluster.

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.

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

Actions Performed By DBCA for Oracle RAC Databases

Review this information to understand about DBCA actions during Oracle RAC database creation.

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.

Using DBCA to Create an Oracle RAC One Node Database

Configure Oracle RAC One Node database using Oracle DBCA after creating a software-only setup of Oracle RAC software on cluster nodes.

After installation of Oracle Real Application Clusters (Oracle RAC) software, start Oracle Database Configuration Assistant (Oracle DBCA).

1. From the Database Operation page, select the option **Create Database**.
2. On the Creation Mode page, select **Advanced Configuration**.
3. On the Deployment Type 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.

   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.

Installing the Oracle Database Vault Option

Installing and configuring Oracle Database Vault requires actions during and after installation.

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

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

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.

> ⓘ **See Also**
>
> *Oracle Database Vault Administrator's Guide* for more information about default Oracle Database Vault configuration

Configuring Oracle Database Vault Using Oracle DBCA

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

Perform the following steps to install Oracle Database Vault using Oracle Database Configuration Assistant (Oracle 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

Deleting an Oracle RAC Database Using Oracle DBCA

Delete an Oracle RAC database using Oracle Database Configuration Assistant (Oracle DBCA) with all of the database components.

Deleting an Oracle Real Application Clusters (Oracle RAC) database using Oracle 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.

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

Oracle Real Application Clusters Postinstallation Procedures

Review this information to complete the postinstallation tasks after you have installed Oracle Real Application Clusters (Oracle RAC).

> ⓘ **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.

- [Required Postinstallation Tasks](#)
  Download and apply required patches for your software release after completing your initial installation.

- [Recommended Postinstallation Tasks](#)
  Oracle recommends that you complete these tasks after installation.

- [Product-Specific Postinstallation Tasks](#)
  You must complete postinstallation tasks before you can use some Oracle Database features.

- [Enabling and Disabling Oracle Database Options After Installation](#)
  The $chopt$ tool changes your database options after installation.

Required Postinstallation Tasks

Download and apply required patches for your software release after completing your initial installation.

- [Downloading Release Update Patches](#)
  Download and install Release Updates (RU) and Monthly Recommended Patches (MRPs) patches for your Oracle software after you complete installation.

- [Setting External Jobs Ownership for HP-UX Installations](#)
  On HP-UX platforms only, set external jobs ownership to a low-privilege user.

- [Setting the Oracle User Environment Variables](#)
  Set the environment variables in the installation owner user profile file on each cluster node.

- [Recompile Invalid Objects in the Database](#)
  After you install, patch, or upgrade a database, recompile invalid objects on the CDB and PDBs with a recompilation driver script.

- [Configuring Services on Oracle RAC and Oracle RAC One Node Databases](#)
  You must add services to the PDBs after installation, if you add PDBs after installation.

- **Copying Oracle ASM Password File For Oracle RAC One Node Database**
  Copy the Oracle ASM password file, if configured, to all candidate nodes configured for fail over after installing Oracle RAC One Node database.

Downloading Release Update Patches

Download and install Release Updates (RU) and Monthly Recommended Patches (MRPs) patches for your Oracle software after you complete installation.

Oracle provides quarterly updates in the form of Release Updates (RU) and Monthly Recommended Patches (MRPs). Oracle no longer releases patch sets. For more information, see My Oracle Support Note 2285040.1.

Check the My Oracle Support website for required updates for your installation.

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

   https://support.oracle.com

2. In the **sign in to other Oracle Support portals** section, click **Support for Oracle Hardware, Software, and Managed Cloud services** to 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. In the **Patch Search** region, select **Product or Family (Advanced)**.

5. On the **Product or Family (Advanced)** display, provide information about the product, release, and platform for which you want to obtain patches, and click **Search**.

   The Patch Search pane opens, displaying the results of your search.

6. Select the patch number and click **ReadMe**.

   The README page is displayed. It contains information about the patch and how to apply the patches to your installation.

7. Uncompress the Oracle patch updates that you downloaded from My Oracle Support.

Related Topics

- My Oracle Support note 888.1
- Patch Delivery Methods for Oracle Database

Setting External Jobs Ownership for HP-UX Installations

On HP-UX platforms only, set external jobs ownership to a low-privilege user.

Complete the following procedure to set external jobs ownership to the low-privilege user extjob:

1. Log in as root.

2. Go to the Oracle Database Oracle home directory:

   # cd $ORACLE_HOME/rdbms/admin/

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

Modify externaljob.ora only as root.

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.

Setting the Oracle User Environment Variables

Set the environment variables in the installation owner user profile file on each cluster node.

You need to set ORACLE_BASE and ORACLE_HOME, and ORACLE_SID; also add ORACLE_HOME/bin to the path environment variable.

- Set environment variables as shown, in the following example:

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/21.0.0/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.

Recompile Invalid Objects in the Database

After you install, patch, or upgrade a database, recompile invalid objects on the CDB and PDBs with a recompilation driver script.

By default, AutoUpgrade performs a recompilation of invalid Oracle objects, which is controlled by the configuraiton file run_utlrp local parameter (default: prefix.run_utlrp=yes). In addition, Oracle provides the recompilation scripts utlrp.sql and utlprp.sql. These scripts are located in the Oracle_home/rdbms/admin directory.

> ⓘ **Note**
>
> Starting with AutoUpgrade 23.1, when you run the AutoUpgrade utility, AutoUpgrade runs the utlprpom.sql script, and does not run utlrp.sql. When AutoUpgrade is used for upgrades to Oracle Database 12c Release 2 (12.2.0.1) and later releases, AutoUpgrade only recompiles invalid objects owned by Oracle-maintained schemas. Because database upgrades do not need to touch user objects, AutoUpgrade maintains this policy when it recompiles invalid objects.

After installing a database, recomplile all invalid objects;

1. Change directory to Oracle_home/rdbms/admin. For example

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

2. Use the catcon.pl script in the Oracle home to run utlrp.sql. For example:

$ORACLE_HOME/perl/bin/perl catcon.pl --n 1 --e --b utlrp --d "'.'" utlrp.sql

Note the following conditions of this use case:

- --n parameter: is set to 1, so the script runs each PDB recompilation in sequence.
- --e parameter: turns echo on.
- --b parameter: Sets the log file base name. It is set to utlrp.

Expect a time delay for the serial recompilation of PDBs to complete. Depending on the number of PDBs that you are upgrading, the recompilation can extend significantly beyond the time required for the upgrade scripts to complete.

The utlrp.sql script automatically recompiles invalid objects in either serial or parallel recompilation, based on both the number of invalid objects, and on the number of CPUs available. CPUs are calculated using the number of CPUs (*cpu_count*) multiplied by the number of threads for each CPU (*parallel_threads_per_cpu*). On Oracle Real Application Clusters (Oracle RAC), this number is added across all Oracle RAC nodes.

After patching or upgrading a database, there is more than one approach you can use to recompile invalid Oracle-owned and user-owned objects:

Recompile all invalid objects (the invalid objects in both Oracle and user schemas) by using utlrp.sql or utlprp.sql.

If time is a factor and the type of invalid objects is predominately application owned, then you can recompile Oracle-owned invalid objects first, and defer recompiling application-owned invalid objects to a later time. To recompile invalid objects in Oracle schemas, use utlprpom.sql. To recompile the remaining invalid objects, use utlrp.sql or utlprp.sql.

> ⓘ **Note**
>
> When you use either utlprp.sql or utlprpom.sql, note that both scripts require you to define the degree of parallelism that the script should use, or determine the number of parallel recompile jobs to use.

The script uses syntax as follows, where *base* is the base name you want to have given to log files, *N* is the number of PDBs on which you want to run recompilation jobs in parallel (degrees of parallelism), *script.sql* is the Oracle recompilation script you chose to use, and *P* is the number of PDBs on which you want to run in parallel:

$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -b *base* -d $ORACLE_HOME/rdbms/admin
    -n *N* -l /tmp *script.sql* '--p*P*'

Suppose you are running recompilation in a CDB using the log file base name recomp, with a degrees of parallelism setting of 3 jobs per PDB container, the script you choose to use is utlprp.sql, and you want to recompile across at most 10 PDBs at a time. In that case, the syntax you use to run the recompile operation is similar to the following,

$ORACLE_HOME/perl/bin/perl $ORACLE_HOME/rdbms/admin/catcon.pl -b recomp -d $ORACLE_HOME/rdbms/admin -n 10 -l /tmp utlprp.sql '--p3'

Related Topics

- Syntax and Parameters for catcon.pl

Configuring Services on Oracle RAC and Oracle RAC One Node Databases

You must add services to the PDBs after installation, if you add PDBs after installation.

If you do not add services to pluggable databases (PDBs), and then the Oracle RAC One Node multitenant container database (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.

- Use the following srvctl command syntax to add services to a PDB, 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

Copying Oracle ASM Password File For Oracle RAC One Node Database

Copy the Oracle ASM password file, if configured, to all candidate nodes configured for fail over after installing Oracle RAC One Node database.

If you install Oracle RAC One Node database using Oracle Database Configuration Assistant (Oracle DBCA) and do not use shared password files stored in Oracle ASM, then you need not perform this task since DBCA copies the Oracle ASM password file to a shared location.

> ⓘ **See Also**
>
> *Oracle Real Application Clusters Administration and Deployment Guide* for information about administering Oracle RAC One Node

Recommended Postinstallation Tasks

Oracle recommends that you complete these tasks after installation.

- **Setting Up Additional User Accounts**
  You can set up additional user accounts to manage your database.

- **About Installing Oracle Autonomous Health Framework**
  Oracle Autonomous Health Framework is pre-installed with Oracle Database.

Setting 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*.

About Installing Oracle Autonomous Health Framework

Oracle Autonomous Health Framework is pre-installed with Oracle Database.

Oracle Autonomous Health Framework performs proactive heath checks and collects diagnostics data for the Oracle software stack. Oracle Autonomous Health Framework updates are available on My Oracle Support and also shipped with Oracle Database Release Updates.

Oracle recommends that you update to the latest version of Oracle Autonomous Health Framework by either applying the latest Database Release Updates or by downloading and installing it from My Oracle Support Note 2550798.1:

https://support.oracle.com/epmos/faces/DocContentDisplay?id=2550798.1&parent=DOCUMENTATION&sourceId=USERGUIDE

Oracle Autonomous Health Framework includes the functionality from Oracle ORAchk, Oracle EXAchk, and Oracle Trace File Analyzer (TFA). Oracle Autonomous Health Framework extends health check coverage to the entire Oracle software stack, based on critical and reoccurring problems. Oracle Autonomous Health Framework proactively scans for known problems with Oracle products and deployments, including the following:

- Single-instance Oracle Database

- Oracle Grid Infrastructure

- Oracle Real Application Clusters

- Maximum Availability Architecture (MAA) Validation

- Upgrade Readiness Validations

- Oracle GoldenGate

- Oracle Exadata

- Oracle SuperCluster

- Oracle Zero Data Loss Recovery Appliance

Related Topics

- *Oracle Autonomous Health Framework Checks and Diagnostics User's Guide*

Product-Specific Postinstallation Tasks

You must complete postinstallation tasks before you can use some Oracle Database features.

- [Configuring Oracle Database Vault](#)
  You must register Oracle Database Vault with Oracle RAC database and create database user accounts before you can use Oracle Database Vault feature.

- [Configuring Oracle Database Security Settings](#)
  Use Oracle Database Configuration Assistant (Oracle DBCA) commands to change security settings after installation.

- [Configuring Oracle Label Security](#)
  You must configure Oracle Label Security after Oracle RAC database installation to use Oracle Label Security feature.

- [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.

- [Configuring Storage for External Tables, Shared Files, or Directory Objects](#)
  Store external tables, shared files, or directory objects on shared storage that is accessible to all cluster nodes.

Configuring Oracle Database Vault

You must register Oracle Database Vault with Oracle RAC database and create database user accounts before you can use Oracle Database Vault feature.

Oracle Universal Installer (OUI) installs Oracle Database Vault by default when you install the Oracle Real Application Clusters (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.

Configuring Oracle Database Security Settings

Use Oracle Database Configuration Assistant (Oracle DBCA) commands to change security settings after installation.

To enable or disable the database security configuration after installation, you must use command-line Oracle Database Configuration Assistant (Oracle DBCA) options. By design, the Oracle DBCA graphical user interface (GUI) does not have the option to enable or disable secure configuration.

- Enable the security settings after installation, use a command of the following form, where myRACdb1.example.com is the either the name of the local database instance, or the value set for the initialization parameter DB_UNIQUE_NAME:

  $ $ORACLE_HOME/bin/dbca –configureDatabase –sourceDB myRACdb1.example.com -SID –enableSecurityConfiguration true

Configuring Oracle Label Security

You must configure Oracle Label Security after Oracle RAC database installation to use Oracle Label Security feature.

You can configure Oracle Label Security in two ways: with Oracle Internet Directory integration, or without Oracle Internet Directory integration.

Table 4-1   Options to Configure Oracle Label Security After Installation

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 then configure the OLS with Oracle Internet Directory integration option.

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

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

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

Store external tables, shared files, or directory objects on shared storage that is accessible to all cluster nodes.

If your Oracle RAC database uses files that are external to the database, then locate the external files 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 Advanced 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 and Upgrade Guide for Linux* for more information about configuring storage.

Enabling and Disabling Oracle Database Options After Installation

The chopt tool changes your 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.

- [Chopt Tool](#)
  Use the chopt tool after installation to add or remove Oracle Database options.

Chopt Tool

Use the chopt tool after installation to add or remove Oracle Database options.

Purpose

The chopt tool is a command-line utility that enables and disables database options.

> ⓘ **Note**
>
> If you install or clone an Oracle Database image, then all Oracle Database options are enabled by default.

Prerequisites

You must complete installation before you can use the chopt tool.

File Path

The tool is located in the *ORACLE_HOME*/bin directory

Syntax

chopt [enable | disable] *db_option*

Options

Command Option	Description
olap	Oracle OLAP
partitioning	Oracle Partitioning
rat	Oracle Real Application Testing

> ⓘ **Note**
>
> - When you enable or disable OLAP, you must run the SYS.XOQ_VALIDATE and SYS.APS_VALIDATE procedures to update the database registry. When you disable OLAP, its status in the database registry should be OPTION OFF and when you enable OLAP, its status in the database registry should be VALID.
>
> - The Oracle Advanced Analytics (OAA) feature is enabled by default for Oracle Database. You cannot disable it using the chopt tool.

Examples

To use the chopt tool to modify your Oracle Database, you must shut down the database before you run the chopt tool, and then start up the database after you add or remove database options.

Example 4-1   Enabling Oracle Real Application Testing Using chopt

The following example shows how to use the chopt tool to enable the Oracle Real Application Testing option in an Oracle Database called Sales:

```
cd $ORACLE_HOME/bin
srvctl stop database -d Sales
chopt enable rat
srvctl start database -d Sales
```

5

Using Server Pools with Oracle RAC

Understand the server pool concepts in Oracle Real Application Clusters (Oracle RAC) environments.

- **Policy-Managed Clusters and Server Pools**
  Use policy-based management and server pools features of Oracle Clusterware for effective management of Oracle database and applications.

- **Oracle RAC Database and Server Pools**
  Oracle RAC databases support two types of server pool management styles and deployment models.

- **About Creating Server Pools for Oracle RAC Databases**
  You can create server pools either before deploying Oracle RAC database software or while creating an Oracle RAC database using Oracle DBCA.

- **Oracle RAC One Node and Server Pools**
  Review the following information about Oracle RAC One Node and server pools.

Policy-Managed Clusters and Server Pools

Use policy-based management and server pools features of Oracle Clusterware for effective management of Oracle database and applications.

- **About Policy-Managed Clusters and Capacity Management**
  Oracle Clusterware uses policy-based management of servers and resources used by Oracle databases or applications.

- **About Server Pools**
  When Oracle Clusterware is installed, two server pools are created automatically: *Generic* and *Free*.

- **Server Pools and Server Categorization**
  Manage servers using server pools by identifying servers distinguished by particular attributes

- **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.

- **How Server Pools Work**
  Server pools divide the cluster into groups of servers hosting singleton and uniform database services and applications.

About Policy-Managed Clusters and Capacity Management

Oracle Clusterware uses policy-based management of servers and resources used by Oracle databases or applications.

Oracle Clusterware has 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, 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.

Starting with Oracle Grid Infrastructure 21c, policy-managed databases are deprecated.

You can continue to use existing server pools, and create new pools and policies. Resources using existing server pools can continue to use them transparently.

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

About 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.

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
* IMPORTANCE and ACL can be edited

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

Server Pools and Server Categorization

Manage servers using server pools by identifying servers distinguished by particular attributes

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.

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:

Starting with Oracle Grid Infrastructure 21c, policy-managed databases are deprecated.

You can continue to use existing server pools, and create new pools and policies. Resources using existing server pools can continue to use them transparently.

The use of CRS configuration policies and the CRS policy set can be desupported in a future release. In place of server pools and policy-managed databases, Oracle recommends that you use the new "Merged" management style.

- 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

How Server Pools Work

Server pools divide the cluster into groups of servers hosting singleton and uniform database services and applications.

Server pools 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

Oracle RAC Database and Server Pools

Oracle RAC databases support two types of server pool 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.

  Starting with Oracle Grid Infrastructure 21c, policy-managed databases are deprecated. You can continue to use existing server pools, and create new pools and policies. Resources using existing server pools can continue to use them transparently.

- **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

About Creating Server Pools for Oracle RAC Databases

You can create server pools either before deploying Oracle RAC database software or while creating an Oracle RAC database using Oracle DBCA.

Oracle recommends that you create server pools before you deploy database software and databases, instead of creating server pools using Oracle Database Configuration Assistant (Oracle DBCA) while creating an Oracle RAC database. 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

Oracle RAC One Node and Server Pools

Review the following information 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.

6
Understanding the Oracle RAC Installed Configuration

Understand the groups and environment settings for the Oracle Real Application Clusters (Oracle RAC) installed configuration.

- [Understanding the Configured Environment in Oracle RAC](#)
  Oracle NETCA and Oracle DBCA configure your environment to meet the requirements for database creation and Oracle Enterprise Manager discovery of the databases.

- [Understanding Operating System Privileges Groups](#)
  Review this information for system privileges required for Oracle Database or Oracle Automatic Storage Management (Oracle ASM) administration.

- [Understanding Time Zone Settings on Cluster Nodes](#)
  Oracle RAC requires that all cluster nodes have the same time zone setting.

- [Understanding the Server Parameter File for Oracle RAC](#)
  Oracle Database creates an SPFILE, during database creation, either in an Oracle ASM disk group or a cluster file system in the location that you specify.

- [About ORATAB Configuration for Oracle RAC](#)
  Oracle creates an entry for each Oracle RAC database in the oratab configuration file.

- [Database Components Created Using Oracle DBCA](#)
  Understand the database components that Oracle Database Configuration Assistant (Oracle DBCA) creates.

- [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.

- [About Initialization Parameter Files](#)
  Oracle recommends using the server parameter file (SPFILE) for storing Oracle Database initialization parameters.

- [Oracle Net Services Configuration for Oracle RAC Databases](#)
  You can access an Oracle RAC database using a client and server configuration or through one or more middle tiers, with or without connection pooling.

- [Performance Features of Oracle Net Services and Oracle RAC](#)
  Oracle RAC databases provide the important benefits of connection load balancing and failover.

- [Oracle Net Services Configuration Files and Parameters](#)
  Review this information to understand the Oracle Net Services configuration files and parameters for an Oracle RAC database.

Understanding the Configured Environment in Oracle RAC

Oracle NETCA and Oracle DBCA configure your environment to meet the requirements for database creation and Oracle Enterprise Manager discovery of the databases.

> ⓘ **Note**
>
> Oracle Net Configuration Assistant (NETCA) and Database Configuration Assistant (DBCA) create configuration files on each node in your cluster.

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.

Understanding Operating System Privileges Groups

Review this information for system privileges required for Oracle Database or Oracle Automatic Storage Management (Oracle ASM) administration.

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 must 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 the following table:

Table 6-1   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.
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.
OSRACDBA	racdba	SYSRAC privileges to perform day to day administration of Oracle databases on an Oracle RAC cluster. 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

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 srvctl setenv database -env 'TZ=*time zone*'

Understanding the Server Parameter File for Oracle RAC

Oracle Database creates an SPFILE, during database creation, either in an Oracle ASM disk group or a cluster file system in the location that you specify.

All instances in the cluster database use the same SPFILE at startup. Do not directly edit the SPFILE with an editor because the SPFILE is a binary file. 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

About 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.

Oracle ASM Agent creates Oracle ASM oratab entries if the node is pinned in the CSS. 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, if your system meets any of the following conditions:

*   The database is a single instance Oracle Database

*   The database is an admin-managed Oracle RAC database

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 must 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

Database Components Created Using Oracle DBCA

Understand the database components that Oracle Database Configuration Assistant (Oracle DBCA) creates.

- About 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.

- About Control Files
  The database is configured with two control files that must be stored on shared storage.

- About Online Redo Log Files
  Each database instance must have at least two online redo log files.

About 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. The following table shows the tablespace names used by an Oracle RAC database and the types of data they contain.

Table 6-2   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.
UNDOTBSn	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.

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.

About 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*

About 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

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

About 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

Oracle Net Services Configuration for Oracle RAC Databases

You can access an Oracle RAC database using a client and 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.

- [Database Services for an Oracle RAC Database](#)
  Each database is represented by one or more services and a service is identified by a service name. A client uses service names to identify databases.

- [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.

- [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.

- [Understanding SCANs](#)
  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).

- [About Connecting to an Oracle RAC Database Using SCANs](#)
  Oracle recommends that you configure Oracle RAC database clients to use the SCAN to connect to the database instead of configuring the tnsnames.ora file.

- [About Listener Configuration for an Oracle RAC Database](#)
  An Oracle Database receives connection requests through the local listener.

- [About Service Registration for an Oracle RAC Database](#)
  An Oracle Database service automatically registers with the listeners specified in the database initialization parameters LOCAL_LISTENER and REMOTE_LISTENER.

- [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.

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

Database Services for an Oracle RAC Database

Each database is represented by one or more services and a service is identified by a service name. A client uses service names to identify databases.

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. Create additional services for client and application connections to the database.

A service name, for example sales.example.com, 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.

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.

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 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.

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.

Oracle recommends against using the easy connect method with SCAN as it provides no high availability support.

Understanding SCANs

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).

- About the SCAN
  During the installation of Oracle Grid Infrastructure, several Oracle Clusterware resources are created for the SCAN.

- About 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.

- About 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.

About the SCAN

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 Oracle Single Client Access Name (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. 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.

Related Topics

• *Oracle Grid Infrastructure Installation and Upgrade Guide*

About 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 11g 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

About 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. The SCAN listener also supports HTTP protocol for communication with Oracle XML Database (XDB).

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

About Connecting to an Oracle RAC Database Using SCANs

Oracle recommends that you configure Oracle RAC database clients to use the SCAN 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 11g 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.

If the SCAN is resolved by DNS, then DNS returns all three SCAN VIP addresses 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 to the client.

Oracle Database 21c database clients use SCAN to connect to the database. Oracle recommends against using the easy connect method with SCAN because the easy connect method does not have the ability to specify timeouts and retries for connection establishment. Instead, applications must use an Oracle Net connect descriptor with the following format:

```
(DESCRIPTION =
    (CONNECT_TIMEOUT=90) (RETRY_COUNT=20)(RETRY_DELAY=3)
(TRANSPORT_CONNECT_TIMEOUT=3)
        ( ADDRESS = (PROTOCOL = TCP)(HOST=scan)(PORT=1521))
          (CONNECT_DATA=(SERVICE_NAME=service_name)))
```

scan represents the SCAN for your cluster. If you do not specify a port number, then the default value of 1521 is used for the TCP port identifier. The *service_name* is the name of a dynamic database service.

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.

Example 6-1   Connecting to Oracle RAC Using an Oracle Net Connect Descriptor

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:

```
(DESCRIPTION =
    (CONNECT_TIMEOUT=90) (RETRY_COUNT=20)(RETRY_DELAY=3)
(TRANSPORT_CONNECT_TIMEOUT=3)
        ( ADDRESS = (PROTOCOL = TCP)(HOST=sales1-scan.mycluster.example.com)(PORT=1521))
            (CONNECT_DATA=(SERVICE_NAME=oltp.example.com)))
```

If the SCAN is resolved by DNS, then DNS returns all three SCAN VIP addresses 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 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.

About 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 11g 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.

> ⚠ **Caution**
>
> Do not modify the REMOTE_REGISTRATION_ADDRESS parameter for your Oracle RAC deployment. The REMOTE_REGISTRATION_ADDRESS parameter is set by default to protect against TNS poisoning. Refer to *Oracle Database Net Services Reference* for more information about REMOTE_REGISTRATION_ADDRESS parameter.

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).

About Service Registration for an Oracle RAC Database

An Oracle 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

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 Load Balancing Actions for Oracle RAC Connections That Use SCAN:

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 6-1   Load Balancing Actions for Oracle RAC Connections That Use SCAN
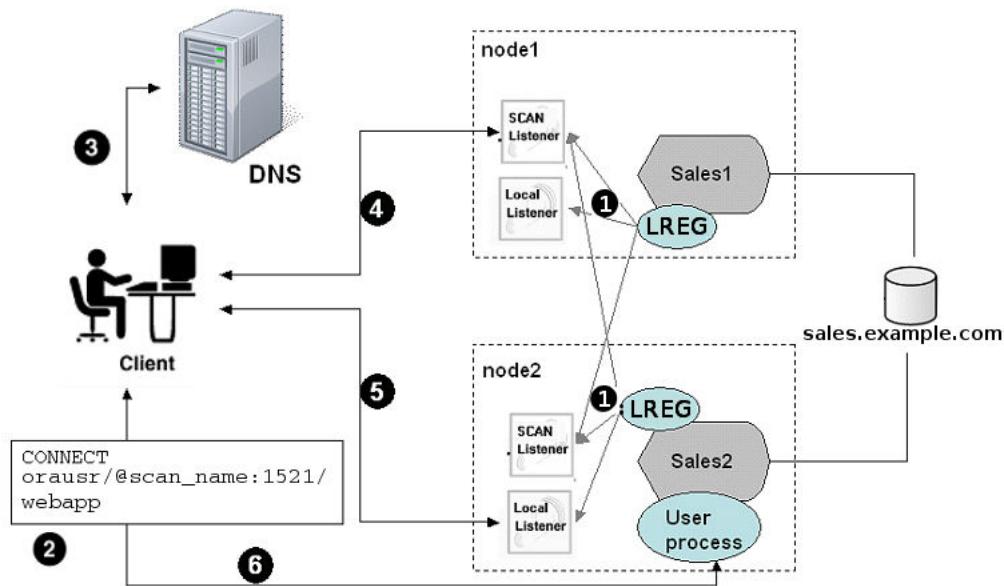
Performance Features of Oracle Net Services and Oracle RAC

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

- Load Balancing of Connections to Oracle RAC Databases
  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.

- Connection Failover for Oracle RAC Databases
  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.

- Shared Server Configuration for an Oracle RAC Database
  Standalone Oracle databases perform load balancing by distributing connections among the shared server dispatcher processes.

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 and Upgrade Guide for Linux* 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

Connection Failover for Oracle RAC Databases

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.

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.

If a node fails, then the service connecting to the VIP is relocated transparently to a remaining 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 remaining node.

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

Oracle Net Services Configuration Files and Parameters

Review this information to understand the Oracle Net Services configuration files and parameters for an Oracle RAC database.

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.

- **[Database Initialization Parameters for Database Service Registration](#)**
  An Oracle Database service automatically registers with the listeners specified in the LOCAL_LISTENER and REMOTE_LISTENER parameters.

- **[Net Service Names and the tnsnames.ora File](#)**
  The Oracle RAC database installation process creates a tnsnames.ora file on each node. This file acts as a repository of net service names.

- **[Net Service Names Created by Oracle DBCA](#)**
  Oracle Database Configuration Assistant (Oracle DBCA) creates net service names for database connections and instance connections.

- **[Listener Configuration and the listener.ora File](#)**
  Oracle recommends that you let the Oracle Agent manage Oracle listeners for Oracle Databases in an Oracle RAC environment.

- **[Net Services Profile File (sqlnet.ora)](#)**
  Use Oracle NETCA to create the Oracle Net Services profile, or the sqlnet.ora file.

Database Initialization Parameters for Database Service Registration

An Oracle 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

Net Service Names and the tnsnames.ora File

The Oracle RAC database 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 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. However, you can create the file in other directories. Oracle Net checks for the tnsnames.ora file in the following order:

1. The directory specified by the TNS_ADMIN environment variable.

2. If the TNS_ADMIN environment variable is not set or the file is not found in the TNS_ADMIN directory, then Oracle Net checks the *Grid_home*/network/admin and ORACLE_HOME/network/admin directories. For a read-only Oracle home, Oracle Net checks the ORACLE_BASE_HOME/network/admin directory.

3. For a read-only Oracle home, if the file is not available in the ORACLE_BASE_HOME/network/admin directory, then Oracle Net checks for the file in the ORACLE_HOME/network/admin directory.

The order for checking tnsnames.ora is the same for both the Oracle home and Oracle Grid Infrastructure home (Grid home) installations.

With Oracle Clusterware 11g 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';
```

Related Topics

- Oracle Database Administrator's Guide

- Oracle Database Net Services Administrator's Guide

Net Service Names Created by Oracle DBCA

Oracle Database Configuration Assistant (Oracle DBCA) creates net service names for database connections and instance connections.

- [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.

- [Net Service Names for Instance Connections](#)
  Clients that connect to a particular instance of the database use the net service name for the instance.

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.

When you use Oracle Database Configuration Assistant (Oracle DBCA) to create an Oracle RAC database, the Oracle DBCA creates a database service that has the same name as the database. 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. Clients that use this database service can connect to any database instance for the Oracle RAC database. 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.

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.

Example 6-2   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.

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 6-3   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)
   )
 )
```

Listener Configuration and the listener.ora File

Oracle recommends that you let the Oracle Agent manage Oracle listeners for Oracle Databases in an Oracle RAC environment.

> ⓘ **Note**
>
> If you enable GNS, then you do not have to manually configure the listener.

*   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.

*   Remote Listeners for an Oracle RAC Database
    A **remote listener**, such as SCAN listener, is a listener residing on one computer that redirects connections to a database instance on another computer.

*   Managing Multiple Listeners for an Oracle RAC Database
    Review to understand how to use SRVCTL and TNS_ADMIN to manage listeners

*   How Oracle Database Uses the Listener File (listener.ora)
    The listener.ora file is the configuration file for a listener, which includes protocol addresses, list of services, and control parameters.

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.

Related Topics

* [Net Service Names and the tnsnames.ora File](#)
  The Oracle RAC database installation process creates a tnsnames.ora file on each node. This file acts as a repository of net service names.

> ⓘ **See Also**
>
> * *Net Service Names and the tnsnames.ora* 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

Remote Listeners for an Oracle RAC Database

A **remote listener**, such as SCAN listener, is a listener residing on one computer that redirects connections to a database instance on another computer.

> ⚠ **Caution**
>
> Do not modify the REMOTE_REGISTRATION_ADDRESS parameter for your Oracle RAC deployment. The REMOTE_REGISTRATION_ADDRESS parameter is set by default to protect against TNS poisoning. Refer to *Oracle Database Net Services Reference* for more information about REMOTE_REGISTRATION_ADDRESS parameter.

In Oracle RAC environments, Oracle recommends that you let the Oracle Agent manage the Oracle listeners for the databases.

Related Topics

* [Net Service Names and the tnsnames.ora File](#)
  The Oracle RAC database installation process creates a tnsnames.ora file on each node. This file acts as a repository of net service names.

> ⓘ **See Also**
>
> * *Net Service Names and the tnsnames.ora* 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

Managing Multiple Listeners for an Oracle RAC Database

Review to understand how to use SRVCTL and TNS_ADMIN to manage listeners

To administer Oracle Database 21c 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 21c.

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.

How Oracle Database Uses the Listener File (listener.ora)

The listener.ora file is the configuration file for a listener, which includes protocol addresses, list of services, and control parameters.

The listener.ora file 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 11g 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 6-4   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/21.0.0/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
```

Net Services Profile File (sqlnet.ora)

Use Oracle NETCA to create 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.

Oracle Net Configuration Assistant (NETCA) creates the following entry in the sqlnet.ora file:

```
NAMES.DIRECTORY_PATH=(TNSNAMES, EZCONNECT)
```

The parameter NAMES.DIRECTORY_PATH specifies the priority order of the naming methods to use to resolve connect identifiers to connect descriptors.

> ⓘ **See Also**
>
> • *Oracle Database Net Services Administrator's Guide* for more information about the sqlnet.ora file

7
Removing Oracle Database Software

Use the deinstall command that is included in Oracle homes to remove Oracle software. Oracle does not support the removal of individual products or components.

> ⚠️ **Caution**
>
> If you have a single-instance database on a node in a cluster, and if you have multiple databases with the same global database name (GDN), then you cannot use the deinstall command to remove one database only.

- **About Oracle Deinstallation Options**
  You can stop and remove Oracle Database software and components in an Oracle Database home with the deinstall command.

- **Identifying All Instances On a Cluster**
  Review this information to identify all instances associated with the Oracle home you want to remove.

- **Oracle Deinstallation (Deinstall)**
  You can run the deinstall command from an Oracle home directory after installation.

- **Deinstallation Examples for Oracle Database**
  Use these examples to help you understand how to run the deinstall command.

- **Deinstallation Parameter File Example for Oracle RAC**
  You can run the deinstall command on Oracle RAC Databases with the -paramfile option to use the values you specify in the parameter file.

About Oracle Deinstallation Options

You can stop and remove Oracle Database software and components in an Oracle Database home with the deinstall command.

You can remove the following software using deinstall :

- Oracle Database
- Oracle Grid Infrastructure, which includes Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM)
- Oracle Real Application Clusters (Oracle RAC)
- Grid Infrastructure Management Repository (GIMR)
- Oracle Database Client

The deinstall command is available in Oracle home directories after installation. It is located in the $ORACLE_HOME/deinstall directory.

deinstall creates a response file by using information in the Oracle home and using the information you provide. You can use a response file that you generated previously by running the deinstall command using the -checkonly option. You can also edit the response file template.

If you run deinstall to remove an Oracle Grid Infrastructure installation, then the deinstaller prompts you to run the deinstall command as the root user. For Oracle Grid Infrastructure for a cluster, the script is rootcrs.sh, and for Oracle Grid Infrastructure for a standalone server (Oracle Restart), the script is roothas.sh.

> ⓘ **Note**
>
> You must run the deinstall command from the same release to remove Oracle software. Do not run the deinstall command from a later release to remove Oracle software from an earlier release. For example, do not run the deinstall command from the 21c Oracle home to remove Oracle software from an existing 19c Oracle home.

If the software in the Oracle home is not running (for example, after an unsuccessful installation), then deinstall cannot determine the configuration, and you must provide all the configuration details either interactively or in a response file.

In addition, before you run deinstall for Oracle Grid Infrastructure installations:

- Dismount Oracle Advanced Cluster File System (Oracle ACFS) and disable Oracle Automatic Storage Management Dynamic Volume Manager (Oracle ADVM).

- If Grid Naming Service (GNS) is in use, then notify your DNS administrator to delete the subdomain entry from the DNS.

Files Deleted by deinstall

When you run deinstall, if the central inventory (oraInventory) contains no other registered homes besides the home that you are deconfiguring and removing, then deinstall 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 deinstall deletes this data.

> ⚠ **Caution**
>
> deinstall deletes Oracle Database configuration files, user data, and fast recovery area (FRA) files even if they are located outside of the Oracle base directory path.

Identifying All Instances On a Cluster

Review this information to identify all instances associated with the Oracle home you want to remove.

1. Enter the following command to identify all instances associated with the Oracle home you want to remove, where *dbname* is the name of the database:

   $ srvctl status database -d *dbname*

   The output of this command contains entries similar to the following:

   +ASM1:/u01/app/21.0.0/grid:N
   CUST:/u01/app/oracle/product/21.0.0/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/21.0.0/grid) and the CUST Oracle database instance are associated with the Oracle home directory /u01/app/oracle/product/21.0.0/dbhome_1.

2. 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

Oracle Deinstallation (Deinstall)

You can run the deinstall command from an Oracle home directory after installation.

Purpose

deinstall stops Oracle software, and removes Oracle software and configuration files on the operating system for a specific Oracle home.

Syntax

The deinstall command uses the following syntax:

(./deinstall [-silent] [-checkonly] [-paramfile *complete path of input response file*]
[-params name1=value name2=*value* . . .]
[-o *complete path of directory for saving files*]
[-tmpdir *complete path of temporary directory to use*]
[-logdir *complete path of log directory to use*] [-local] [-skipLocalHomeDeletion] [-skipRemoteHomeDeletion] [-help]

Parameters

Parameter	Description
-silent	Use this flag to run deinstall in noninteractive mode. This option requires one of the following: • A working system that it can access to determine the installation and configuration information. The -silent flag does not work with failed installations. • A response file that contains the configuration values for the Oracle home that is being deinstalled or deconfigured. You can generate a response file to use or modify by running deinstall with the -checkonly flag. deinstall then discovers information from the Oracle home to deinstall and deconfigure. It generates the response file that 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 directory.
-checkonly	Use this flag to check the status of the Oracle software home configuration. Running deinstall with the -checkonly flag does not remove the Oracle configuration. The -checkonly flag generates a response file that you can then use with the deinstall command and -silent option.
-paramfile *complete path of input response file*	Use this flag to run deinstall with a response file in a location other than the default. When you use this flag, provide the complete path where the response file is located. The default location of the response file is $ORACLE_HOME/deinstall/response.
-params [*name1=value name2=value name3=value . . .*]	Use this flag with a response file to override one or more values to change in a response 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 response file (deinstall.rsp.tmpl) is saved. The default location of the response file is $ORACLE_HOME/deinstall/response .
-tmpdir *complete path of temporary directory to use*	Use this flag to specify a non-default location where deinstall writes the temporary files for the deinstallation.
-logdir *complete path of log directory to use*	Use this flag to specify a non-default location where deinstall writes the log files for the deinstallation.

Parameter	Description
-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.
-skipLocalHomeDeletion	Use this flag in Oracle Grid Infrastructure installations on a multinode environment to deconfigure a local Grid home without deleting the Grid home.
-skipRemoteHomeDeletion	Use this flag in Oracle Grid Infrastructure installations on a multinode environment to deconfigure a remote Grid home without deleting the Grid home.
-help	Use this option to obtain additional information about the command option flags.

Deinstallation Examples for Oracle Database

Use these examples to help you understand how to run the deinstall command.

Run deinstall from the $ORACLE_HOME/deinstall directory. The deinstallation starts without prompting you for the Oracle home path.

$ ./deinstall

You can generate a deinstallation response file by running deinstall with the -checkonly flag. Alternatively, you can use the response file template located at $ORACLE_HOME/deinstall/response/deinstall.rsp.tmpl. If you have a response file, then use the optional flag -paramfile to provide a path to the response file.

In the following example, the deinstall command is in the path/u01/app/oracle/product/21.0.0/dbhome_1/deinstall. It uses a response file called my_db_paramfile.tmpl in the software owner location /home/usr/oracle:

$ cd /u01/app/oracle/product/21.0.0/dbhome_1/deinstall
$ ./deinstall -paramfile /home/usr/oracle/my_db_paramfile.tmpl

To remove the Oracle Grid Infrastructure home, use the deinstall command in the Oracle Grid Infrastructure home.

In this example, the Oracle Grid Infrastructure home is /u01/app/21.0.0/grid

$ cd /u01/app/21.0.0/grid/deinstall
$ ./deinstall -paramfile /home/usr/oracle/my_grid_paramfile.tmpl

Deinstallation Parameter File Example for Oracle RAC

You can run the deinstall command on Oracle RAC Databases with the -paramfile option to use the values you specify in the parameter file.

Deinstallation Parameter File Description and Example

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/21.0.0/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 deinstall command from) is myserver, and the OSDBA group is dba:

```
#Copyright (c) 2005, 2014 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/21.0.0/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/21.0.0/dbhome_1/
SPFILE_LOC.sidb=
```

8

Using Scripts or Response Files to Create Oracle RAC Databases

Review this information 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.

- [Using DBCA to Generate Installation Scripts for Oracle RAC](#)
  You can use Oracle DBCA to generate scripts to create an Oracle RAC database, create a database using the generated scripts, and prepare the database for use.

- [About DBCA Noninteractive (Silent) Configuration for Oracle RAC](#)
  Use Oracle DBCA to perform a noninteractive, or silent configuration of Oracle RAC database using the templates supplied by Oracle or custom templates.

- [Using Oracle DBCA Commands for Noninteractive (Silent) Configuration for Oracle RAC](#)
  Use `createDatabase` command to create an Oracle RAC database using Oracle DBCA.

- [How Response Files Work](#)
  Response files can assist you with installing an Oracle product multiple times on multiple computers.

- [Preparing Response Files](#)
  Review this information to prepare response files for use during silent mode or response file mode installations.

- [Running Oracle Universal Installer Using a Response File](#)
  After creating the response file, run Oracle Universal Installer at the command line, specifying the response file you created, to perform the installation.

- [Postinstallation Configuration Using Response File Created During Installation](#)
  Use response files to configure Oracle software after installation. You can use the same response file created during installation to also complete postinstallation configuration.

- [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.

Using DBCA to Generate Installation Scripts for Oracle RAC

You can use Oracle DBCA to generate scripts to create an Oracle RAC database, create a database using the generated scripts, and prepare the database for use.

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, note the path name for use in the next step.

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 SPFILE for each instance.

5. Configure Oracle Net Services to support your new database and instances.

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 the following document:

   Oracle Real Application Clusters Administration and Deployment Guide

Related Topics

- [Understanding the Configured Environment in Oracle RAC](#)
  Oracle NETCA and Oracle DBCA configure your environment to meet the requirements for database creation and Oracle Enterprise Manager discovery of the databases.

About DBCA Noninteractive (Silent) Configuration for Oracle RAC

Use Oracle DBCA to perform a noninteractive, or silent configuration of Oracle RAC database using the templates supplied by Oracle or custom templates.

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.

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.

Using Oracle DBCA Commands for Noninteractive (Silent) Configuration for Oracle RAC

Use createDatabase command to create an Oracle RAC database using Oracle DBCA.

- Use the following command syntax to create an Oracle Real Application Clusters (Oracle RAC) database using the general purpose template and AUTOMATIC management policy, placing the data files in an existing Oracle ASM disk group, configuring a TDE wallet, and defining default encryption algorithm during the database creation. It also configures Oracle Machine Learning for Python in the database and sets the SYS, SYSTEM, and DBSNMP passwords to *password*, which is the password for each account:

```
# su oracle -c "$ORACLE_HOME/bin/dbca -silent -createDatabase -templateName General_Purpose.dbc
-gdbName $DBNAME -sid $ORACLE_SID -managementPolicy AUTOMATIC -configureTDE true -
tdeAlgorithm AES256 -encryptTablespaces ALL -sysPassword password -systemPassword password
-tdeWalletPassword password [-tdeWalletLoginType AUTO_LOGIN] -dbsnmpPassword password -
emConfiguration LOCAL
-storageType ASM -diskGroupName ASMgrp1 -datafileJarLocation $ORACLE_HOME/assistants/dbca/
templates -nodeinfo node1,node2 -characterset "WE8ISO8859P1"
-configureoml4py -oml4pyConfigTablespace SYSAUX -enableOml4pyEmbeddedExecution true
```

> ⓘ **Note**
>
> The Oracle Machine Learning for Python feature is supported only on Linux operating systems.

Nodes node1 and node2 are the cluster nodes on which Oracle RAC database instances are created with a TDE wallet configuration and AES256 as the default encryption algorithm for tablespace creation operations. 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.

> ⓘ **See Also**
>
> *Oracle Database Administrator's Guide* for a complete description of Oracle Database Configuration Assistant (Oracle DBCA) commands and options

How Response Files Work

Response files can assist you with installing an Oracle product multiple times on multiple computers.

- Modes for Oracle DBCA Noninteractive (Silent) Configuration
  You run the Oracle DBCA installer from a command prompt using either silent mode or response file mode.

- Reasons for Using Silent Mode or Response File Mode
  Review this section for use cases for running the installer in silent mode or response file mode.

- 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.

- Using Response Files
  Use response files to install and configure Oracle products using the installer in silent or response file mode.

Modes for Oracle DBCA Noninteractive (Silent) Configuration

You run the Oracle DBCA installer from a command prompt using either silent mode or response file mode.

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, provide the Oracle home path for the ORACLE_HOME environment variable:

ORACLE_HOME=/u01/app/oracle/product/21.0.0/dbhome_1

Reasons for Using Silent Mode or Response File Mode

Review this section for 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.

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.

Using Response Files

Use response files 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. Prepare a response file.

2. Run the installer in silent or response file mode.

3. Run the root scripts as prompted by Oracle Universal Installer.

4. If you completed a software-only installation, then run Net Configuration Assistant and Oracle DBCA in silent or response file mode to create the database listener and an Oracle Database instance respectively.

Preparing Response Files

Review this information to prepare response files for use during silent mode or response file mode installations.

- [Editing a Response File Template](#)
  Oracle provides response file templates for each product and each configuration tool.

- [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.

Editing a Response File Template

Oracle provides response file templates for each product and each configuration tool.

About Response File Templates

For Oracle Database, the response file templates are located in the $ORACLE_HOME/install/response directory. For Oracle Grid Infrastructure, the response file templates are located in the *Grid_home*/install/response directory.

Where, *Grid_home* is the Oracle Grid Infrastructure home directory path.

> ⓘ **Note**
>
> If you copied the software to a hard disk, then the response files are located in the $ORACLE_HOME/install/response directory.

All response file templates contain comment entries, sample formats, examples, and other useful instructions. Read the response file instructions to understand how to specify values for the response file variables, so that you can customize your installation.

The following table lists the response files provided with this software:

Table 8-1   Response Files for Oracle Database and Oracle Grid Infrastructure

Response File	Description
db_install.rsp	Silent installation of Oracle Database.
dbca.rsp	Silent creation and configuration of Oracle Database using Oracle DBCA.
netca.rsp	Silent configuration of Oracle Net using Oracle NETCA.
gridsetup.rsp	Silent configuration of Oracle Grid Infrastructure installations.

> ⚠ **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:

   For example, for Oracle Database:

   $ cp $ORACLE_HOME/install/response/db_install.rsp *local_directory*

2. Open the response file in a text editor:

$ vi /*local_directory*/db_install.rsp

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. Also, ensure that your response file name has the .rsp suffix.

4. Secure the response file by changing the permissions on the file to 600:

$ chmod 600 /*local_dir*/db_install.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).

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.

Recording response files 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 set up 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
>
> OUI does not save passwords while recording the 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 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.

> ⓘ **Note**
>
> Ensure that your response file name has the .rsp suffix.

5. 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.

Running Oracle Universal Installer Using a Response File

After creating the response file, run Oracle Universal Installer at the command line, specifying the response file you created, to perform the installation.

Run Oracle Universal Installer at the command line, specifying the response file you created. The Oracle Universal Installer executables, runInstaller and gridSetup.sh, provide several options. For help information on the full set of these options, run the gridSetup.sh or runInstaller command with the -help option. For example:

- For Oracle Database:

  $ $ORACLE_HOME/runInstaller -help

- For Oracle Grid Infrastructure:

  $ /u01/app/21.0.0/grid/gridSetup.sh -help

The help information appears in a window after some time.

Perform the following steps to run the installer using a response file:

1. Complete the preinstallation tasks for a normal installation.

2. Log in as the software installation owner user.

3. If you are completing a response file mode installation, then set the operating system DISPLAY environment variable for the user running the installation.

> ⓘ **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:

   - For Oracle Database:

     $ $ORACLE_HOME/runInstaller [-silent] -responseFile *responsefilename*

- For Oracle Grid Infrastructure:

$ /u01/app/21.0.0/grid/gridSetup.sh [-silent] -responseFile *responsefilename*

> **ⓘ Note**
>
> Do not specify a relative path to the response file. If you specify a relative path, then the installer fails.

In this example:

- -silent runs the installer in silent mode.
- *responsefilename* is the full path and file name of the installation response file that you configured.

5. If this is the first time you are installing Oracle software on your system, then Oracle Universal Installer prompts you to run the orainstRoot.sh script. Log in as the root user and run the orainstRoot.sh script:

```
$ su root
password:
# /u01/app/oraInventory/orainstRoot.sh
```

> **ⓘ Note**
>
> You do not have to manually create the oraInst.loc file. Running the orainstRoot.sh script is sufficient as it specifies the location of the Oracle Inventory directory.

6. When the installation completes, log in as the root user and run the root.sh script. For example:

```
$ su root
password:
# $ORACLE_HOME/root.sh
```

Postinstallation Configuration Using Response File Created During Installation

Use response files to configure Oracle software after installation. You can use the same response file created during installation to also complete postinstallation configuration.

- [Using the Installation Response File for Postinstallation Configuration](#)
  You can use the response file created during installation to also complete postinstallation configuration.
- [Running Postinstallation Configuration Using Response File](#)
  You can use a response file to complete postinstallation tasks on one or more servers simultaneously.

Using the Installation Response File for Postinstallation Configuration

You can use the response file created during installation to also complete postinstallation configuration.

Run the installer with the -executeConfigTools option to configure configuration assistants after installing Oracle Grid Infrastructure or Oracle Database. You can use the response file located at $ORACLE_HOME/install/response/*product_timestamp.rsp* to obtain the passwords required to run the configuration tools. You must update the response file with the required passwords before running the -executeConfigTools command.

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.

Example 8-1   Response File Passwords for Oracle Grid Infrastructure (grid user)

grid.install.crs.config.ipmi.bmcPassword=*password*
grid.install.asm.SYSASMPassword=*password*
grid.install.asm.monitorPassword=*password*
grid.install.config.emAdminPassword=*password*

If you do not have a BMC card, or you do not want to enable IPMI, then leave the ipmi.bmcPassword input field blank.

If you do not want to enable Oracle Enterprise Manager for management, then leave the emAdminPassword password field blank.

Example 8-2   Response File Passwords for Oracle Grid Infrastructure for a Standalone Server (oracle user)

oracle.install.asm.SYSASMPassword=*password*
oracle.install.asm.monitorPassword=*password*
oracle.install.config.emAdminPassword=*password*

If you do not want to enable Oracle Enterprise Manager for management, then leave the emAdminPassword password field blank.

Example 8-3   Response File Passwords for Oracle Database (oracle user)

This example illustrates the passwords to specify for use with the database configuration assistants.

oracle.install.db.config.starterdb.password.SYS=*password*
oracle.install.db.config.starterdb.password.SYSTEM=*password*
oracle.install.db.config.starterdb.password.DBSNMP=*password*
oracle.install.db.config.starterdb.password.PDBADMIN=*password*
oracle.install.db.config.starterdb.emAdminPassword=*password*
oracle.install.db.config.asm.ASMSNMPPassword=*password*

You can also specify oracle.install.db.config.starterdb.password.ALL=*password* to use the same password for all database users.

The database configuration assistants require the SYS, SYSTEM, PDBADMIN, and DBSNMP passwords for use with Oracle DBCA. You must specify the following passwords, depending on your system configuration:

- If the database uses Oracle Automatic Storage Management (Oracle ASM) for storage, then you must specify a password for the ASMSNMPPassword variable. If you are not using Oracle ASM, then leave the value for this password variable blank.

Running Postinstallation Configuration Using Response File

You can use a response file to complete postinstallation tasks on one or more servers simultaneously.

Complete this procedure to run configuration assistants with the executeConfigTools command and a response file.

1. Edit the response file and specify the required passwords for your configuration. You can use the response file created during installation, located at $ORACLE_HOME/install/response/*product_timestamp.rsp*.

2. Change directory to the Oracle home containing the installation software. For example:

   For Oracle Grid Infrastructure:

   cd *Grid_home*

   Where, *Grid_home* is the path to the Oracle Grid Infrastructure home directory /u01/app/21.0.0/grid

   For Oracle Database:

   cd $ORACLE_HOME

3. Run the configuration script using the following syntax:

   For Oracle Grid Infrastructure:

   $ ./gridSetup.sh -executeConfigTools -responseFile *Grid_home*/install/response/*product_timestamp.rsp*

   For Oracle Database:

   $ ./runInstaller -executeConfigTools -responseFile $ORACLE_HOME/install/response/*product_timestamp.rsp*

   For Oracle Database, you can also run the response file located in the directory $ORACLE_HOME/install/response/:

   $ ./runInstaller -executeConfigTools -responseFile $ORACLE_HOME/install/response/*db_install.rsp*

   The postinstallation configuration tool runs the installer in the graphical user interface mode, displaying the progress of the postinstallation configuration. Specify the [-silent] option to run the postinstallation configuration in the silent mode.

For example, for Oracle Grid Infrastructure:

$ ./gridSetup.sh -executeConfigTools -responseFile /u01/app/21.0.0/grid/install/response/
grid_2019-01-09_01-03-36PM.rsp [-silent]

For Oracle Database:

$ ./runInstaller -executeConfigTools -responseFile /u01/app/oracle/product/21.0.0/dbhome_1/response/
db_2019-01-09_01-03-36PM.rsp [-silent]

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.

> ⓘ **Note**
>
> If you copied the software to a hard disk, then the response file template is located in the /response directory.

- [Running Oracle DBCA Using Response Files](#)
  You can run Oracle DBCA in response file mode to configure and start an Oracle database on the system.
- [Running Net Configuration Assistant Using Response Files](#)
  You can run NETCA in silent mode to configure and start an Oracle Net Listener on the system, and configure naming methods and Oracle Net service names.

Running Oracle DBCA Using Response Files

You can run Oracle DBCA in response file mode to configure and start an Oracle database on the system.

To run Oracle Database Configuration Assistant (Oracle DBCA) 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. To run Oracle DBCA in response file mode, you must use the -responseFile flag in combination with the -silent flag. You must also use a graphical display and set the DISPLAY environment variable.

To run Oracle DBCA in response file mode:

1. Copy the dbca.rsp response file template from the response file directory to a directory on your system:

   $ cp $ORACLE_HOME/assistants/dbca/dbca.rsp /local_directory

   In this example, directory_path is the path of the directory where you have copied the installation binaries.

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 Oracle 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. Follow the instructions in the file to edit the file.

> ⓘ **Note**
>
> Oracle 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 Oracle DBCA in response file mode, set the DISPLAY environment variable.

6. Use the following command syntax to run Oracle DBCA in silent or response file mode using a response file:

$ORACLE_HOME/bin/dbca [-silent] -createDatabase -responseFile /local_dir/dbca.rsp

In this example:

- -silent option indicates that Oracle DBCA runs in silent mode.

- local_dir is the full path of the directory where you copied the dbca.rsp response file template.

During configuration, Oracle DBCA displays a window that contains the status messages and a progress bar.

Running Net Configuration Assistant Using Response Files

You can run NETCA in silent mode to configure and start an Oracle Net Listener on the system, and configure naming methods and Oracle Net service names.

To run Oracle Net Configuration Assistant (NETCA) in silent mode, you must copy and edit a response file template. Oracle provides a response file template named netca.rsp in the $ORACLE_HOME/assistants/netca 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 $ORACLE_HOME/assistants/netca/netca.rsp local_directory

2. Open the response file in a text editor:

$ vi /local_directory/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_directory*/netca.rsp

In this command:

- The /silent option indicates to run Net Configuration Assistant in silent mode.

- *local_directory* is the full path of the directory where you copied the netca.rsp response file template.

9

Directory Structure for Oracle RAC Environments

Understand the directory structure for Oracle Real Application Clusters (Oracle RAC) software environments.

- Understanding the Oracle RAC Directory Structure
  When you install Oracle RAC database, all subdirectories except for the Oracle Inventory directory, are placed under a top-level Oracle base directory.
- Directory Structures for Oracle RAC
  Understand the hierarchical directory tree of a sample Optimal Flexible Architecture (OFA)-compliant database for Oracle RAC.

Understanding the Oracle RAC Directory Structure

When you install Oracle RAC database, all subdirectories except for the Oracle Inventory directory, are placed under a top-level Oracle base directory.

The Oracle Automatic Storage Management (Oracle ASM) home (if applicable), and the Oracle Clusterware home, are placed 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 Linux* for your platform for more information about the Oracle home and admin directories

Directory Structures for Oracle RAC

Understand the hierarchical directory tree of a sample Optimal Flexible Architecture (OFA)-compliant database for Oracle RAC.

Table 9-1   Directory Structure for A Sample OFA-Compliant Environment

Directory	Description
$ORACLE_BASE	The default ORACLE_BASE directory, where the software owner is the oracle user. For example: /u01/app/oracle
installation_type	Type of installation under the Oracle base directory. For example, when installing Oracle Database 21c, the value for installation type is product/21.0.0/dbhome_1. For example: /u01/app/oracle/product/21.0.0/dbhome_1

Table 9-1   (Cont.) Directory Structure for A Sample OFA-Compliant Environment

Directory	Description
$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 21c software, then the path is as follows:
	/u01/app/oracle/product/21.0.0/dbhome_2
	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	The administrative directory. Note that with Oracle Database 11g, bdump, cdump, and udump files are relocated to the directory associated with ADR_BASE. For example:
	/u01/app/oracle/admin
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	The dump destinations for database server.
/pfile	
$ADR_BASE/bdump	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.
$ADR_BASE/cdump	
$ADR_BASE/udump	
	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/21.0.0/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

10

Preparing to Upgrade an Existing Oracle RAC Database

Understand how you can prepare an Oracle Real Application Clusters (Oracle RAC) database for patch updates or upgrade.

- [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.

- [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.

- [AutoUpgrade and Cluster-Managed Oracle Database Upgrades](#)
  Starting with Oracle Database 21c, AutoUpgrade can automate Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node upgrades using the multitenant architecture.

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.

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.

- [cluvfy Command Options for Database Upgrade Validation](#)

- [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.

cluvfy Command Options for Database Upgrade Validation

Use the Cluster Verification Utility (CVU) command `cluvfy stage -pre dbinst -upgrade` to check the readiness of your Oracle RAC installation for upgrades.

Purpose

Running `cluvfy` with the `-predbinst -upgrade` options performs system checks to confirm if the cluster is in a correct state for upgrading from an existing Oracle RAC installation.

Syntax

cluvfy stage -pre dbinst -upgrade -src_dbhome *src_RAChome* [-dbname
db_names_list] -dest_dbhome *dest_RAChome* -dest_version *dest_version*
[-fixup] [-fixupnoexec][-method root|-method sudo -user *user_name*
[-location *dir_path*]] [-verbose]

The command performs the appropriate checks on all the nodes in the cluster before setting up an Oracle RAC Database. The following table describes the options for the command.

Options

Table 10-1   Cluster Verification Utility Command Options for Oracle RAC Databases

Option	Description
-src_dbhome *src_RAChome*	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*	Location of the Oracle RAC home for upgrade, where *dest_RAChome* is the path to the Oracle RAC home.
-dest_version *dest_version*	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: 21.0.0.0.0.
—user *user_name*	User name to access all the nodes with root privileges.
-location *dir_path*	Full file system path for the sudo executable.
-dbname *db_names_list*	List of unique names of the databases being upgraded.
-verbose	Displays detailed output of individual checks.
-fixup	If specified, on verification failure, performs fixup operations.
-fixupnoexec	If specified, on verification failure, fix up data is generated and the instruction for manual execution of the generated fix ups is displayed.

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/19.0.0/dbhome_1
-dest_dbhome /u01/app/oracle/product/21.0.0/dbhome_1
-dest_version 21.0.0.0.0 -fixup -verbose

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*

AutoUpgrade and Cluster-Managed Oracle Database Upgrades

Starting with Oracle Database 21c, AutoUpgrade can automate Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node upgrades using the multitenant architecture.

When you run Autoupgrade on an Oracle Database instance that is managed with Oracle Clusterware, and that database instance is a pluggable database (PDB) on a container database (CDB), AutoUpgrade detects how the database is configured:

- Oracle Real Application Clusters (Oracle RAC) cluster member node, using Oracle Automatic Storage Management (Oracle ASM), or a supported shared file system.

- Oracle Real Application Clusters One Node (Oracle RAC One Node) using Oracle ASM

Based on that detection, AutoUpgrade proceeds with the upgrade steps required for that configuration type.

> ⓘ **Note**
>
> Choosing this upgrade option requires downtime of the cluster while AutoUpgrade completes upgrades of database instances, and system configuration. If you use Oracle Enterprise Manager, then you must reconfigure it after the upgrade.

Overview of Steps for Cluster-Managed Oracle Database Instances

When you have installed the new release binaries, and AutoUpgrade detects cluster-managed databases, Autoupgrade completes the following steps in sequence:

1. Disables Oracle RAC services (if needed).

2. Disables Oracle RAC or Oracle RAC One Node databases managed by Oracle Clusterware.

3. Shuts down all Oracle Database instances in the cluster.

4. Starts the local instance, with the cluster initialization parameter $CLUSTER\_DATABASE=FALSE$.

5. Upgrades the local Oracle RAC or Oracle RAC One Node database instance.

6. Registers and enables Oracle RAC in target version in the clusterware ($CLUSTER\_DATABASE=TRUE$).

7. Starts all Oracle Database instances on the cluster.

Requirements for Using AutoUpgrade to Upgrade Oracle Databases Managed with Oracle Clusterware

Your source release Oracle Database instances and your target release Oracle Database instances must meet the following requirements:

- You must address any issues that are identified by AutoUpgrade pre-upgrade validation checks.

- The source release Oracle Database instances that you want to upgrade must be supported for direct upgrade to the target Oracle Database release.

- The source Oracle Database release and the target Oracle Database release must be running on an operating system supported for Oracle Database on both the source and target releases. If your source database is running on an operating system that is not supported with the target Oracle Database release, then you must migrate your source Oracle Database to a supported operating system before upgrade.

- The source release Oracle home must be installed on all cluster member nodes, and configured as an Oracle RAC or Oracle RAC One Node database.

- The target release Oracle home must be installed and configured as an Oracle RAC or Oracle RAC One Node database

- The source Oracle Database must be up and running.

- The operating system must be either Oracle Linux (formerly Oracle Enterprise Linux), or a supported Portable Operating System Interface POSIX Linux or Unix operating system.

- The user account performing the upgrade must be configured with operating system authentication.

- The system uses Oracle Automatic Storage Management for storage (recommended), or a supported shared storage system.

- The system must have a properly configured and functioning Oracle Grid Infrastructure software configuration for Oracle RAC or Oracle RAC One Node.

Required Tasks for Database Administrators to Use AutoUpgrade

As the database administrator, you must complete the following tasks:

- Create an adequate backup strategy to prevent data loss from any problems resulting from the upgrade.

- Configure Listener and Transparent Network Substrate (TNS) files, both for local tnsnames.ora and SCAN listeners, if needed.

- Configure Oracle Wallet certificates and management (if needed). The Oracle Wallet must be set for automatic login.

Related Topics

- Enabling Full Deployments for AutoUpgrade

11
Configuring Oracle Homes

Understand how read-only Oracle homes work and how you can configure read-only Oracle homes.

- **Evolution of Oracle Homes**
  Learn about read-only Oracle home concepts like Oracle base home, Oracle base config, and orabasetab.

- **Copying demo Directories to Oracle Base Home**
  In a read-only mode ORACLE_HOME, you must copy the demo directories listed in this topic from ORACLE_HOME to ORACLE_BASE_HOME.

- **File Path and Directory Changes in Read-Only Oracle Homes**
  Learn from these examples the differences between hierarchical file mappings in a read-only Oracle home compared to a read/write Oracle home.

Evolution of Oracle Homes

Learn about read-only Oracle home concepts like Oracle base home, Oracle base config, and orabasetab.

- **About Read-Only Oracle Homes**
  In Oracle Database 21c, an Oracle Database installation configures all Oracle Database homes in read-only mode by default. A read-only Oracle home does not imply that the file system and the Oracle home is in read-only mode. The file system and mount point should always be in read/write mode.

- **About Oracle Base Home**
  In a read-only ORACLE_HOME the user-specific files, instance-specific files, and log files reside in a location known as the ORACLE_BASE_HOME.

- **About Oracle Base Config**
  In a read-only ORACLE_HOME the configuration files reside in a location known as ORACLE_BASE_CONFIG.

- **About orabasetab**
  The orabasetab file is used to define fundamental directories based on $ORACLE_HOME, ORACLE_BASE, ORACLE_BASE_HOME and ORACLE_BASE_CONFIG.

- **Location of Networking Files in a Read-Only Oracle Home**
  Learn about the search order and location of networking files like sqlnet.ora, tnsnames.ora, or listener.ora in a read-only Oracle home.

- **Location of Networking Files in a Read-Only Oracle Home**
  Learn about the search order and location of networking files like sqlnet.ora, tnsnames.ora, or listener.ora in a read-only Oracle home.

About Read-Only Oracle Homes

In Oracle Database 21c, an Oracle Database installation configures all Oracle Database homes in read-only mode by default. A read-only Oracle home does not imply that the file

system and the Oracle home is in read-only mode. The file system and mount point should always be in read/write mode.

A read-only Oracle Home simplifies provisioning by implementing separation of installation and configuration.

Before Oracle Database 21c, the default ORACLE_HOME layout combined ORACLE_HOME, ORACLE_BASE_HOME and ORACLE_BASE_CONFIG into a single location. Starting with Oracle Database 21c, the only available configuration is a read-only ORACLE_HOME where ORACLE_BASE_HOME and ORACLE_BASE_CONFIG are located separately from ORACLE_HOME.

In a read-only Oracle home, all the configuration data and log files reside outside of the read-only Oracle home. See, [File Path and Directory Changes in Read-Only Oracle Homes](#).

Apart from the traditional ORACLE_BASE and ORACLE_HOME directories, the following directories contain files that used to be in ORACLE_HOME:

• ORACLE_BASE_HOME

• ORACLE_BASE_CONFIG

Ensure that you enable read-only Oracle home before you create the database or listener.

When you run the roohctl script to enable a read-only Oracle home, the Oracle home file system will continue to be a read/write file system.

> ⓘ **Note**
>
> This feature does not affect how database administrators monitor, diagnose, and tune their system performance.

About Oracle Base Home

In a read-only ORACLE_HOME the user-specific files, instance-specific files, and log files reside in a location known as the ORACLE_BASE_HOME.

Starting with Oracle Database 21c, the only available configuration is a read-only ORACLE_HOME where ORACLE_BASE_HOME and ORACLE_BASE_CONFIG are located separately from ORACLE_HOME.

In a read-only ORACLE_HOME, the ORACLE_BASE_HOME directory is not co-located with ORACLE_HOME but is located at ORACLE_BASE/homes/HOME_NAME. Where, HOME_NAME is the internal name for ORACLE_HOME.

For example, the networking directories network/admin, network/trace, and network/log are located in the ORACLE_BASE_HOME directory. In a read-only ORACLE_HOME the networking directories are located in ORACLE_BASE/homes/HOME_NAME.

To print the ORACLE_BASE_HOME path, run the orabasehome command from the $ORACLE_HOME/bin directory:

```
$ setenv ORACLE_HOME /u01/app/oracle/product/21.0.0/dbhome_1
$ cd $ORACLE_HOME/bin
$ ./orabasehome
```

For example:

$ ./orabasehome
/u01/app/oracle/homes/OraDB21Home1

Where, /u01/app/oracle is ORACLE_BASE and OraDB21Home1 is HOME_NAME

About Oracle Base Config

In a read-only ORACLE_HOME the configuration files reside in a location known as ORACLE_BASE_CONFIG.

Starting with Oracle Database 21c, the only available configuration is a read-only ORACLE_HOME where ORACLE_BASE_HOME and ORACLE_BASE_CONFIG are located separately from ORACLE_HOME.

In a read-only ORACLE_HOME, the ORACLE_BASE_CONFIG path is the same as ORACLE_BASE.

ORACLE_BASE_CONFIG/dbs contains the configuration files for ORACLE_HOME. Each file in the dbs directory contains $ORACLE_SID so that the directory can be shared by many different ORACLE_SIDs.

To print the ORACLE_BASE_CONFIG path, run the orabaseconfig command from the $ORACLE_HOME/bin directory:

$ setenv ORACLE_HOME /u01/app/oracle/product/21.0.0/dbhome_1
$ cd $ORACLE_HOME/bin
$ ./orabaseconfig

For example:

$ ./orabaseconfig
/u01/app/oracle

Where, /u01/app/oracle is ORACLE_BASE.

About orabasetab

The orabasetab file is used to define fundamental directories based on $ORACLE_HOME, ORACLE_BASE, ORACLE_BASE_HOME and ORACLE_BASE_CONFIG.

Starting with Oracle Database 21c, the only available configuration is a read-only ORACLE_HOME where ORACLE_BASE_HOME and ORACLE_BASE_CONFIG are located separately from ORACLE_HOME.

The orabasetab file resides in ORACLE_HOME/install/orabasetab. To determine if an ORACLE_HOME is read-only, check for the presence of the orabasetab file. The orabasetab file also defines the ORACLE_BASE and the HOME_NAME of the Oracle home. HOME_NAME is the internal name for ORACLE_HOME.

The last line in the orabasetab file, which starts with $ORACLE_HOME, defines the directories for $ORACLE_HOME. The last line consists of four fields, each separate by a colon delimiter(:).

1. The first field matches the current $ORACLE_HOME.

2. The second field defines the ORACLE_BASE for the current ORACLE_HOME.

3. The third field defines the HOME_NAME which is used in constructing the ORACLE_BASE_HOME path in a read-only ORACLE_HOME.

4. The fourth field displays Y in a read-only ORACLE_HOME.

In a read-only ORACLE_HOME, the ORACLE_BASE_HOME path is ORACLE_BASE/homes/HOME_NAME and ORACLE_BASE_CONFIG is the same as ORACLE_BASE.

Viewing an orabasetab File

1. Log in as the Oracle installation owner user account (oracle).

2. Go to the $ORACLE_HOME/install directory.

   ```
   $ cd /u01/app/oracle/product/21.0.0/dbhome_1/install
   ```

3. View the contents of the orabasetab file.

   ```
   $ cat orabasetab
   /u01/app/oracle/product/21.0.0/dbhome_1:/u01/app/oracle:OraDB21Home1:Y:
   ```

   In this example, a Y in the fourth field at the end of the line indicates you have a read-only Oracle home.

Location of Networking Files in a Read-Only Oracle Home

Learn about the search order and location of networking files like sqlnet.ora, tnsnames.ora, or listener.ora in a read-only Oracle home.

The networking directories network/admin, network/trace, and network/log are located in the ORACLE_BASE_HOME directory. In a read/write ORACLE_HOME the networking directories appear to be in ORACLE_HOME because ORACLE_BASE_HOME is co-located with ORACLE_HOME, whereas in a read-only ORACLE_HOME the networking directories are located in ORACLE_BASE/homes/HOME_NAME.

tnsnames.ora

The order for checking the tnsnames.ora file in a read-only Oracle home is:

1. The directory specified by the TNS_ADMIN environment variable.

2. The /etc directory.

3. The ORACLE_BASE_HOME/network/admin directory.

4. The ORACLE_HOME/network/admin directory.

sqlnet.ora

The order for checking the sqlnet.ora file in a read-only Oracle home is:

1. The directory specified by the TNS_ADMIN environment variable.

2. The ORACLE_BASE_HOME/network/admin directory.

3. The ORACLE_HOME/network/admin directory.

Password Files

The search order for password files for Oracle homes is ORACLE_BASE_CONFIG/dbs followed by ORACLE_HOME/dbs. This helps prioritize to search for the password file outside of ORACLE_HOME when read-only Oracle home is enabled.

Related Topics

- Overview of Local Naming Parameters

Location of Networking Files in a Read-Only Oracle Home

Learn about the search order and location of networking files like sqlnet.ora, tnsnames.ora, or listener.ora in a read-only Oracle home.

The networking directories network/admin, network/trace, and network/log are located in the ORACLE_BASE_HOME directory. In a read/write ORACLE_HOME the networking directories appear to be in ORACLE_HOME because ORACLE_BASE_HOME is co-located with ORACLE_HOME, whereas in a read-only ORACLE_HOME the networking directories are located in ORACLE_BASE/homes/HOME_NAME.

tnsnames.ora

The order for checking the tnsnames.ora file in a read-only Oracle home is:

1. The directory specified by the TNS_ADMIN environment variable.

2. The /var/opt/oracle directory.

3. The ORACLE_BASE_HOME/network/admin directory.

4. The ORACLE_HOME/network/admin directory.

sqlnet.ora

The order for checking the sqlnet.ora file in a read-only Oracle home is:

1. The directory specified by the TNS_ADMIN environment variable.

2. The ORACLE_BASE_HOME/network/admin directory.

3. The ORACLE_HOME/network/admin directory.

Password Files

The search order for password files in a read-only Oracle home is ORACLE_BASE_HOME/dbs followed by ORACLE_HOME/dbs.

Related Topics

- Overview of Local Naming Parameters

Copying demo Directories to Oracle Base Home

In a read-only mode ORACLE_HOME, you must copy the demo directories listed in this topic from ORACLE_HOME to ORACLE_BASE_HOME.

Oracle Database contains various demo directories that include a variety of examples and product demonstrations that you can use to learn about the products, options, and features of Oracle Database. In a read-only mode ORACLE_HOME, you cannot use these demo

directories in ORACLE_HOME because writes are performed to these demo directories when they are used.

Many of the demo directories are not available by default. You must install Oracle Database Examples to view and use the examples and product demonstrations.

Copy the respective demo directory to the corresponding location in ORACLE_BASE_HOME. Now, you can use this copy of the demo directory.

You must copy the following demo directories from ORACLE_HOME to ORACLE_BASE_HOME:

- jdbc/demo
- odbc/demo
- ord/http/demo
- precomp/demo
- rdbms/demo
- sqlj/demo
- sqlplus/demo
- xdk/demo

You must also create symbolic links for the odbc/demo, precomp/demo, rdbms/demo, and xdk/demo demo directories. See the "Creating Symbolic Links" section in this topic.

Copying demo Directories

For example, to copy the rdbms/demo directory from ORACLE_HOME to ORACLE_BASE_HOME, perform the following:

1. Login as the Oracle software owner user (oracle).

2. Check if the rdbms/demo directory is copied to ORACLE_BASE_HOME.

   $ ls -l -d $(orabasehome)/rdbms/demo

3. If the rdbms/demo directory has not been copied, then create it and copy it.

   $ mkdir -p $(orabasehome)/rdbms
   $ cp -r $ORACLE_HOME/rdbms/demo $(orabasehome)/rdbms/demo

Similarly, copy all the demo directories listed earlier from ORACLE_HOME to ORACLE_BASE_HOME.

Creating Symbolic Links

You must create symbolic links for the odbc/demo, precomp/demo, rdbms/demo, and xdk/demo demo directories.

For rdbms/demo, replace $ORACLE_HOME/rdbms/demo with a symbolic link to the copy.

1. Ensure that the symbolic link does not already exist.

   $ ls -l -d $ORACLE_HOME/rdbms/demo

2. If $ORACLE_HOME/rdbms/demo is still the original demo directory, rename it and replace it with the symbolic link.

```
$ cd $ORACLE_HOME/rdbms
$ mv demo demo.installed
$ ln -s $(orabasehome)/rdbms/demo $ORACLE_HOME/rdbms/demo
```

For odbc/demo, replace $ORACLE_HOME/odbc/demo with a symbolic link to the copy.

1. Ensure that the symbolic link does not already exist.

```
$ ls -l -d $ORACLE_HOME/odbc/demo
```

2. If $ORACLE_HOME/odbc/demo is still the original demo directory, rename it and replace it with the symbolic link.

```
$ cd $ORACLE_HOME/odbc
$ mv demo demo.installed
$ ln -s $(orabasehome)/odbc/demo $ORACLE_HOME/odbc/demo
```

For precomp/demo, replace $ORACLE_HOME/precomp/demo with a symbolic link to the copy.

1. Ensure that the symbolic link does not already exist.

```
$ ls -l -d $ORACLE_HOME/precomp/demo
```

2. If $ORACLE_HOME/precomp/demo is still the original demo directory, rename it and replace it with the symbolic link.

```
$ cd $ORACLE_HOME/precomp
$ mv demo demo.installed
$ ln -s $(orabasehome)/precomp/demo $ORACLE_HOME/precomp/demo
```

The xdk/demo directory requires a symbolic link at $ORACLE_HOME/xdk/include pointing to $(orabasehome)/xdk/include after you copy the xdk/demo directory.

1. Ensure that the symbolic link does not already exist:

```
$ ls -l -d $ORACLE_HOME/xdk/include
```

2. If the symbolic link does not exist, then, run the following command:

```
$ ln -s $ORACLE_HOME/xdk/include $(orabasehome)/xdk/include
```

> ⓘ **Note**
>
> In the plsql/demo directory, ncmpdemo.sql is unusable in read-only mode.

Copying the init.ora File

Copy the init.ora file from ORACLE_HOME to ORACLE_BASE_HOME.

1. Login as the Oracle software owner user (oracle).

2. Check if the init.ora file exists in ORACLE_BASE_HOME.

   $ ls $(orabasehome)/init.ora

   If an init.ora file exists in ORACLE_BASE_HOME, then update this init.ora file to be in-sync with the $ORACLE_HOME/init.ora file.

3. If the init.ora file does not exist in ORACLE_BASE_HOME, then copy it from ORACLE_HOME.

   $ cp $ORACLE_HOME/init.ora $(orabasehome)/init.ora

Related Topics

• *Oracle Database Examples Installation Guide*

File Path and Directory Changes in Read-Only Oracle Homes

Learn from these examples the differences between hierarchical file mappings in a read-only Oracle home compared to a read/write Oracle home.

Prior to Oracle Database 21c, the default ORACLE_HOME layout combined ORACLE_HOME, ORACLE_BASE_HOME and ORACLE_BASE_CONFIG into a single location. Starting with Oracle Database 21c, the only available configuration is a read-only ORACLE_HOME where ORACLE_BASE_HOME and ORACLE_BASE_CONFIG are located separately from ORACLE_HOME.

This example shows an Optimal Flexible Architecture-compliant Oracle Database installation, for the user oracle, with the ORACLE_HOME, ORACLE_BASE, ORACLE_BASE_HOME, and ORACLE_BASE_CONFIG logical locations. The database files are mounted on /u01 and the HOME_NAME is OraDB21Home1.

This example also shows the changes in the Oracle Database software defined paths of configuration files, log files, and other directories in a read-only Oracle home when compared to a read/write Oracle home.

Table 11-1   Read-Only Oracle Home and Read/Write Oracle Home File Path Examples

Directory	Read-Only Oracle Home File Path (21c onwards)	Read/Write Oracle Home File Path (before 21c)
ORACLE_HOME	/u01/app/oracle/product/21.0.0/dbhome_1	/u01/app/oracle/product/19.0.0/dbhome_1
ORACLE_BASE	/u01/app/oracle/	/u01/app/oracle/
ORACLE_BASE_HOME	ORACLE_BASE/homes/HOME_NAME (or) /u01/app/oracle/homes/OraDB21Home1	ORACLE_HOME (or) /u01/app/oracle/product/19.0.0/dbhome_1
ORACLE_BASE_CONFIG	ORACLE_BASE (or) /u01/app/oracle/	ORACLE_HOME (or) /u01/app/oracle/product/19.0.0/dbhome_1

Table 11-1   (Cont.) Read-Only Oracle Home and Read/Write Oracle Home File Path Examples

Directory	Read-Only Oracle Home File Path (21c onwards)	Read/Write Oracle Home File Path (before 21c)
network	ORACLE_BASE_HOME/network/admin (or) /u01/app/oracle/homes/OraDB21Home1/network/admin	ORACLE_HOME/network/admin (or) /u01/app/oracle/product/19.0.0/dbhome_1/network/admin
dbs	ORACLE_BASE_CONFIG/dbs (or) /u01/app/oracle/dbs	ORACLE_HOME/dbs (or) /u01/app/oracle/product/19.0.0/dbhome_1/dbs
hs	ORACLE_BASE_HOME/hs/admin (or) /u01/app/oracle/homes/OraDB21Home1/hs/admin	ORACLE_HOME/hs/admin (or) /u01/app/oracle/product/19.0.0/dbhome_1/hs/admin
mgw	ORACLE_BASE_HOME/mgw/admin (or) /u01/app/oracle/homes/OraDB21Home1/mgw/admin	ORACLE_HOME/mgw/admin (or) /u01/app/oracle/product/19.0.0/dbhome_1/mgw/admin
drdaas	ORACLE_BASE_HOME/drdaas/admin (or) /u01/app/oracle/homes/OraDB21Home1/drdaas/admin	ORACLE_HOME/drdaas/admin (or) /u01/app/oracle/product/19.0.0/dbhome_1/drdaas/admin

- To print the ORACLE_BASE_HOME path, run:

```
$ setenv ORACLE_HOME /u01/app/oracle/product/21.0.0/dbhome_1
$ cd $ORACLE_HOME/bin
$ ./orabasehome
```

- To print the ORACLE_BASE_CONFIG path, run:

```
$ setenv ORACLE_HOME /u01/app/oracle/product/21.0.0/dbhome_1
$ cd $ORACLE_HOME/bin
$ ./orabaseconfig
```

12

Managing Oracle Database Port Numbers

Review default port numbers. If needed, use these steps to change assigned ports after installation.

- [About Managing Ports](#)
  During installation, Oracle Universal Installer assigns port numbers to components from a set of default port numbers.

- [Port Numbers and Protocols of Oracle Components](#)
  Review this information for 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.

About Managing Ports

During installation, Oracle Universal Installer assigns port numbers to components from a set of default port numbers.

Many Oracle Database 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 host. Enter the following command to identify the ports currently used on your computer:

```
$/bin/netstat -a
```

Most port numbers are assigned during installation. Every component and service has an allotted port range, which is the set of port numbers Oracle Database attempts to use when assigning a port. Oracle Database starts with the lowest number in the range and performs the following checks:

- Is the port used by another Oracle Database installation on the host?

  The installation may be up or down 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 can be any process on the host, even a non-Oracle Database process.

- Is the port listed in the /etc/services file?

If the answer to any of the preceding questions is yes, then Oracle Database moves to the next highest port in the allotted port range, and continues checking until it finds a free port.

Port Numbers and Protocols of Oracle Components

Review this information for 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 12-1   Ports Used in Oracle Components

Component	Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
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 (mDNS)	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
Quality of Management Service (QOMS) Server	The CRS Agent uses port 8888 locally to manage the lifecycle of the container.	8888	8888	TCP	Not applicable
Quality of Management Service (QOMS) Server	Port for the Quality of Management Service server.	8895	8895	RMI	No
Oracle Fleet Patching and Provisioning Server	Port for the Oracle Fleet Patching and Provisioning server.	8889	8889	TCP	Not applicable
Oracle Fleet Patching and Provisioning Server	Port for the Oracle Fleet Patching and Provisioning server.	8896	8896	RMI	No

Table 12-1 (Cont.) Ports Used in Oracle Components

Component	Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
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
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 performs name resolution for the cluster.	53	53	UDP	No
Oracle Grid Naming Service (GNSD)	The Oracle Grid Naming Service daemon performs name resolution for the cluster.	Dynamic	Dynamic	TCP	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 ASM Listener	Port for Oracle ASM Listener.	1525	Port number changes to the next available port. Modifiable manually to any available port.	TCP	No

Table 12-1   (Cont.) Ports Used in Oracle Components

Component	Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
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
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 port 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

Table 12-1   (Cont.) Ports Used in Oracle Components

Component	Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
Oracle Trace File Analyzer (TFA)	The Oracle Trace File Analyzer (TFA) daemons in a cluster communicate securely over ports 5000 to 5005. If the port range is not available on your system, then replace it with the ports available on your system using the tfactl set port command.	5000	5000 to 5005	SSL/TLS	No