

Oracle® Real Application Clusters

Real Application Clusters Installation Guide



19c for Linux and UNIX

E96277-06

April 2022

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Real Application Clusters Real Application Clusters Installation Guide, 19c for Linux and UNIX

E96277-06

Copyright © 2013, 2022, Oracle and/or its affiliates.

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

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Intended Audience	ix
Documentation Accessibility	ix
Set Up Java Access Bridge to Implement Java Accessibility	ix
Related Documentation	x
Conventions	x

1 Oracle RAC Installation Checklist

Deployment Checklist for Oracle RAC Database	1-1
Server Hardware and Software Review Checklist for Oracle RAC Installation	1-2
Installer Planning Checklist for Oracle Database	1-3
Oracle RAC Upgrade Checklist	1-6

2 Installing Oracle RAC and Oracle RAC One Node

About Image-Based Oracle Database Installation	2-1
Setup Wizard Installation Options for Creating Images	2-2
Downloading the Software from Oracle Software Delivery Cloud Portal	2-3
Deciding Between Multitenant Container Databases and Non-CDBs in Oracle RAC	2-3
Installing Oracle RAC and Oracle RAC One Node Databases	2-4
Installing Oracle RAC and Oracle RAC One Node Database Software	2-4
Applying Patches During an Oracle Database Installation or Upgrade	2-6
About Deploying Oracle Databases Using Oracle Fleet Patching and Provisioning	2-7

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

About Oracle Database Configuration Assistant	3-1
Selecting Installation Options for Oracle RAC	3-2
About Cluster Node Selection for Database Installation	3-2
Selecting a Database Name	3-3
Requirements for Database Passwords	3-4

About Automatic Memory Management Installation Options	3-4
About Character Set Selection During Installation	3-5
Managing Database Services After Installation	3-6
Automatic Listener Migration from Earlier Releases	3-7
Verifying Requirements for DBCA	3-7
Tasks to Complete Before Using DBCA to Create any Oracle RAC Database	3-8
Load SSH Keys Into Memory Before Starting DBCA	3-8
Decide on a Naming Convention to Use for Your Oracle RAC Database	3-8
Configure Shared Storage for the Oracle RAC Database	3-9
Selecting DBCA Options to Create an Oracle RAC or Oracle RAC One Node Database	3-9
Starting DBCA	3-10
Cluster Detection and Node Selection when Using DBCA	3-10
Using DBCA to Select Storage to Use With any Oracle RAC Database	3-10
Selecting Server Pool Option with DBCA	3-11
Using DBCA to Specify Database Initialization Parameters for Oracle RAC	3-11
Actions Performed By DBCA for Oracle RAC Databases	3-11
Using DBCA to Create an Oracle RAC One Node Database	3-12
Installing the Oracle Database Vault Option	3-12
Starting the Listener with Oracle Database Vault Installations	3-13
Configuring Oracle Database Vault Using DBCA	3-13
Deleting an Oracle RAC Database Using DBCA	3-14

4 Oracle Real Application Clusters Postinstallation Procedures

Required Postinstallation Tasks	4-1
Downloading Release Update Patches	4-2
Setting External Jobs Ownership for HP-UX Installations	4-3
Setting the Oracle User Environment Variables	4-3
Recompiling All Invalid Objects	4-3
Configuring Services on Oracle RAC and Oracle RAC One Node CDBs	4-4
Copying Oracle ASM Password File For Oracle RAC One Node Database	4-5
Recommended Postinstallation Tasks	4-5
Setting Up Additional User Accounts	4-5
About Installing Oracle Autonomous Health Framework	4-6
Product-Specific Postinstallation Tasks	4-6
Configuring Oracle Database Vault	4-7
Configuring Oracle Database Security Settings	4-7
Configuring Oracle Label Security	4-7
Configuring Oracle XML DB	4-8
Configuring Storage for External Tables, Shared Files, or Directory Objects	4-8
Enabling and Disabling Oracle Database Options After Installation	4-9

5 Using Server Pools with Oracle RAC

Policy-Managed Clusters and Capacity Management	5-1
Server Pools and Server Categorization	5-2
Server Pools and Policy-Based Management	5-2
How Server Pools Work	5-3
About Server Pools	5-3
The Free Server Pool	5-3
The Generic Server Pool	5-3
Oracle RAC Database and Server Pools	5-4
About Creating Server Pools for Oracle RAC Databases	5-5
Oracle RAC One Node and Server Pools	5-6

6 Understanding the Oracle RAC Installed Configuration

Understanding the Configured Environment in Oracle RAC	6-2
Understanding Operating System Privileges Groups	6-2
Understanding Time Zone Settings on Cluster Nodes	6-4
Understanding the Server Parameter File for Oracle RAC	6-4
About ORATAB Configuration for Oracle RAC	6-4
Database Components Created Using Database Configuration Assistant	6-5
About Tablespaces and Data Files	6-5
About Control Files	6-6
About Online Redo Log Files	6-6
Managing Undo Tablespaces in Oracle RAC	6-7
About Initialization Parameter Files	6-7
Oracle Net Services Configuration for Oracle RAC Databases	6-8
Database Services for an Oracle RAC Database	6-9
Naming Methods and Connect Descriptors	6-9
Easy Connect Naming Method	6-9
Understanding SCANS	6-10
About the SCAN	6-10
About SCAN VIP Addresses	6-11
About SCAN Listeners	6-11
About Connecting to an Oracle RAC Database Using SCANS	6-12
About Listener Configuration for an Oracle RAC Database	6-13
About Service Registration for an Oracle RAC Database	6-14
How Database Connections are Created When Using SCANS	6-15
Performance Features of Oracle Net Services and Oracle RAC	6-16

Load Balancing of Connections to Oracle RAC Databases	6-17
Connection Failover for Oracle RAC Databases	6-17
Shared Server Configuration for an Oracle RAC Database	6-17
Oracle Net Services Configuration Files and Parameters	6-18
Database Initialization Parameters for Database Service Registration	6-18
Net Service Names and the tnsnames.ora File	6-19
Net Service Names Created by DBCA	6-20
Net Service Names for Database Connections	6-20
Net Service Names for Instance Connections	6-21
Listener Configuration and the listener.ora File	6-21
Local Listener for an Oracle RAC Database	6-22
Remote Listeners for an Oracle RAC Database	6-23
Managing Multiple Listeners for an Oracle RAC Database	6-24
How Oracle Database Uses the Listener File (listener.ora)	6-24
Net Services Profile File (sqlnet.ora)	6-25

7 Removing Oracle Database Software

About Oracle Deinstallation Options	7-1
Identifying All Instances On a Cluster	7-3
Oracle Deinstallation (Deinstall)	7-3
Deinstallation Examples for Oracle Database	7-5
Deinstallation Parameter File Example for Oracle RAC	7-6

A Using Scripts or Response Files to Create Oracle RAC Databases

Using DBCA to Generate Installation Scripts for Oracle RAC	A-2
About DBCA Noninteractive (Silent) Configuration for Oracle RAC	A-3
Using DBCA Commands for Noninteractive (Silent) Configuration for Oracle RAC	A-3
How Response Files Work	A-4
Reasons for Using Silent Mode or Response File Mode	A-4
Creating a Database Using Oracle ASM for Database Files	A-5
Using Response Files	A-5
Preparing Response Files	A-6
Editing a Response File Template	A-6
Recording Response Files	A-8
Running Oracle Universal Installer Using a Response File	A-9
Postinstallation Configuration Using Response File Created During Installation	A-10
Using the Installation Response File for Postinstallation Configuration	A-11
Running Postinstallation Configuration Using Response File	A-12
Postinstallation Configuration Using the ConfigToolAllCommands Script	A-13

About the Postinstallation Configuration File	A-14
Creating a Password Response File	A-15
Running Postinstallation Configuration Using a Password Response File	A-16
Running Configuration Assistants Using Response Files	A-16
Running Oracle DBCA Using Response Files	A-17
Running Net Configuration Assistant Using Response Files	A-18

B Directory Structure for Oracle RAC Environments

Understanding the Oracle RAC Directory Structure	B-1
Directory Structures for Oracle RAC	B-1

C Preparing to Upgrade an Existing Oracle RAC Database

Backing Up the Oracle RAC Database	C-1
Using CVU to Validate Readiness for Oracle RAC Upgrades	C-1
Using the CVU Database Upgrade Validation Command Options	C-1
Example of Verifying System Upgrade Readiness for Oracle RAC	C-3
Verifying System Readiness for Oracle Database Upgrades	C-3

D Configuring Read-Only Oracle Homes

Evolution of Oracle Homes	D-1
About Read-Only Oracle Homes	D-1
About Oracle Base Homes	D-2
About Oracle Base Config	D-3
About orabasetab	D-3
Enabling a Read-Only Oracle Home	D-4
Copying demo Directories to Oracle Base Home	D-6
Determining if an Oracle Home is Read-Only	D-8
File Path and Directory Changes in Read-Only Oracle Homes	D-9

E Managing Oracle Database Port Numbers

About Managing Ports	E-1
About Viewing Port Numbers and Access URLs	E-2
Port Numbers and Protocols of Oracle Components	E-2

Index

List of Tables

1-1	Deployment Checklist for Oracle RAC Database	1-1
1-2	Hardware and Software Review Checklist for Oracle RAC	1-2
1-3	Oracle Universal Installer Planning Checklist for Oracle Database Installation	1-3
1-4	Oracle RAC Upgrade Checklist	1-7
2-1	Image-Creation Options for Setup Wizard	2-2
4-1	Options to Configure Oracle Label Security After Installation	4-7
6-1	Role-Allocated Oracle System Privileges Operating System Groups	6-3
6-2	Tablespace Names Used with Oracle Real Application Clusters Databases	6-5
A-1	Response Files for Oracle Database and Oracle Grid Infrastructure	A-6
B-1	Directory Structure for A Sample OFA-Compliant Environment	B-1
C-1	Cluster Verification Utility Command Options for Oracle RAC Databases	C-2
D-1	read/write and Read-Only Oracle Home File Path Examples	D-9
E-1	Ports Used in Oracle Components	E-2

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](#)
- [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.
- [Related Documentation](#)
- [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.

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*

Related Documentation

The related documentation for Oracle Database products includes the following manuals:

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Application Express Installation Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Database Concepts*
- *Oracle Database New Features Guide*
- *Oracle Database Licensing Information User Manual*
- *Oracle Database Release Notes*
- *Oracle Database Installation Guide*
- *Oracle Database Examples Installation Guide*
- *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems*
- *Oracle Database Upgrade Guide*
- *Oracle Database 2 Day DBA*
- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX*

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Changes in this Release for Oracle Database

The following are changes in *Oracle Real Application Clusters Installation Guide* for Oracle Database 19c.

- [New Features](#)
Review new features available with Oracle Database 19c.
- [Deprecated Features](#)
Review features that are deprecated starting with Oracle Database 19c.
- [Desupported Features](#)
Review features that are desupported with Oracle Database 19c.
- [Other Changes](#)
Review other changes for Oracle Database 19c.

Related Topics

- [Oracle Database New Features Guide](#)

New Features

Review new features available with Oracle Database 19c.

- [Root Scripts Automation Support for Oracle Database Installation](#)
- [Resupport of Direct File Placement for OCR and Voting Disks](#)
- [Optional Install for the Grid Infrastructure Management Repository](#)

Related Topics

- [Oracle Database New Features Guide](#)

Root Scripts Automation Support for Oracle Database Installation

Starting with Oracle Database 19c, the database installer, or setup wizard, provides options to set up permissions to run the root configuration scripts automatically, as required, during a database installation. You continue to have the option to run the root configuration scripts manually.

Setting up permissions for root configuration scripts to run without user intervention can simplify database installation and help avoid inadvertent permission errors.

Related Topics

- [Installing Oracle RAC and Oracle RAC One Node Database Software](#)
Install Oracle RAC or Oracle RAC One Node software.

Resupport of Direct File Placement for OCR and Voting Disks

Starting with Oracle Grid Infrastructure 19c, the desupport for direct OCR and voting disk file placement on shared file systems is rescinded for Oracle Standalone Clusters. For Oracle Domain Services Clusters the requirement to place OCR and voting files in Oracle Automatic Storage Management (Oracle ASM) on top of files hosted on shared file systems and used as ASM disks remains.

In Oracle Grid Infrastructure 12c Release 2 (12.2), Oracle announced that it would no longer support the placement of the Oracle Grid Infrastructure Oracle Cluster Registry (OCR) and voting files directly on a shared file system. This desupport is now rescinded. Starting with Oracle Grid Infrastructure 19c (19.3), with Oracle Standalone Clusters, you can again place OCR and voting disk files directly on shared file systems.

Related Topics

- Using a Cluster File System for Oracle Clusterware Files

Optional Install for the Grid Infrastructure Management Repository

Starting with Oracle Grid Infrastructure 19c, the Grid Infrastructure Management Repository (GIMR) is optional for new installations of Oracle Standalone Cluster. Oracle Domain Services Clusters still require the installation of a GIMR as a service component.

The data contained in the GIMR is the basis for preventative diagnostics based on applied Machine Learning and can help to increase the availability of Oracle Real Application Clusters (Oracle RAC) databases. Having an optional installation for the GIMR allows for more flexible storage space management and faster deployment, especially during the installation of test and development systems.

Related Topics

- About the Grid Infrastructure Management Repository

Deprecated Features

Review features that are deprecated starting with Oracle Database 19c.

For more information about deprecated features, parameters, and views, refer to *Oracle Database Upgrade Guide*

- **Init.ora Parameter CLUSTER_DATABASE_INSTANCES is Deprecated**

The Oracle Database initialization parameter `CLUSTER_DATABASE_INSTANCES` is deprecated in Oracle Database 19c (19.1)

The `init.ora` parameter `CLUSTER_DATABASE_INSTANCES` specifies the number of configured Oracle Real Application Clusters (Oracle RAC) instances. Starting with Oracle Database 19c and later releases, the number of configurable Oracle RAC instances is derived automatically from the Oracle Clusterware resource definitions. There is no replacement for this parameter, because there is no longer a reason to have this parameter.

- **Deprecation of the SERVICE_NAMES Initialization Parameter**

Starting with Oracle Database 19c, customer use of the `SERVICE_NAMES` parameter is deprecated. It can be desupported in a future release.

The use of the `SERVICE_NAMES` parameter is no longer actively supported. It must not be used for high availability (HA) deployments. It is not supported to use service names parameter for any HA operations. This restriction includes FAN, load balancing, `FAILOVER_TYPE`, `FAILOVER_RESTORE`, `SESSION_STATE_CONSISTENCY`, and any other uses.

To manage your services, Oracle recommends that you use the `SRVCTL` or `GDSCTL` command line utilities, or the `DBMS_SERVICE` package.

Related Topics

- [Oracle Database Upgrade Guide](#)

Desupported Features

Review features that are desupported with Oracle Database 19c.

For more information about desupported features, parameters, and views, refer to *Oracle Database Upgrade Guide*

- **Desupport of Leaf Nodes in Flex Cluster Architecture**

Leaf nodes are no longer supported in the Oracle Flex Cluster Architecture in Oracle Grid Infrastructure 19c.

In Oracle Grid Infrastructure 19c (19.1) and later releases, all nodes in an Oracle Flex Cluster function as hub nodes. The capabilities offered by Leaf nodes in the original implementation of the Oracle Flex Cluster architecture can as easily be served by hub nodes. Therefore, leaf nodes are no longer supported.

- **Desupport of Oracle Real Application Clusters for Standard Edition 2 (SE2) Database Edition**

Starting with Oracle Database 19c, Oracle Real Application Clusters (Oracle RAC) is not supported in Oracle Database Standard Edition 2 (SE2).

Upgrading Oracle Database Standard Edition databases that use Oracle Real Application Clusters (Oracle RAC) functionality from earlier releases to Oracle Database 19c is not possible. To upgrade those databases to Oracle Database 19c, either remove the Oracle RAC functionality before starting the upgrade, or upgrade from Oracle Database Standard Edition to Oracle Database Enterprise Edition. For more information about each step, including how to reconfigure your system after an upgrade, see My Oracle Support Note 2504078.1: "Desupport of Oracle Real Application Clusters (RAC) with Oracle Database Standard Edition 19c."

Related Topics

- [Oracle Database Upgrade Guide](#)
- [My Oracle Support Document 2504878.1](#)

Other Changes

Review other changes for Oracle Database 19c.

- **Rapid Home Provisioning (RHP) Name Change**

Starting with Oracle Database 19c and Oracle Grid Infrastructure 19c, Rapid Home Provisioning is renamed to Fleet Patching and Provisioning (FPP).

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: <ul style="list-style-type: none">• 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: <ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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](#)
Oracle Fleet Patching and Provisioning (Oracle FPP) provides a standardized approach for provisioning, patching and upgrading software homes on any number of nodes from a single cluster.

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

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

Check	Task
Operating System General Requirements	Prepare servers as described in <i>Oracle Grid Infrastructure Installation Guide</i> for your platform.
Review Existing Oracle Installations and Upgrade Plans	<p>To install Oracle RAC 19c, you must have Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) 19c installed on your cluster. The Oracle Clusterware version must be equal to or greater than the Oracle RAC version that you plan to install.</p> <p>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.</p> <p>See Also: <i>Oracle Database Upgrade Guide</i> <i>Oracle Grid Infrastructure Installation Guide</i></p>
Server Hardware on each node	Use identical server hardware on each node, to simplify server maintenance.
Time Zone Requirement	<p>Upgrade the Time Zone File and TSTZ Data:</p> <p>As part of an installation of Oracle Database 19c, time zone version files from 1 to 12 are installed in the following path:</p> <p>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.</p> <p>See <i>Oracle Database Globalization Support Guide</i> for information about how to upgrade the time zone file and TSTZ data</p>
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. Contact your system or storage administrators to have them fix errors manually. CVU is available in the Grid home, in the <code>bin</code> 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	<ul style="list-style-type: none"> 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 19c when you perform an Oracle Grid Infrastructure 19c installation. If you have an existing Oracle home, then you can create a new Oracle home and install Oracle Database 19c 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	<ul style="list-style-type: none"> Ensure an external jobs user is created for HP-UX. Configure a Windows Domain user account to use when installing Oracle RAC on Oracle ASM Cluster File System (Oracle ACFS) on Windows platforms.

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	<p>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/</p> <p>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.</p>

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

Check	Task
Review the Licensing Information	<p>You are permitted to use only those components in the Oracle Database media pack for which you have purchased licenses. For more information about licenses, refer to the following URL:</p> <p><i>Oracle Database Licensing Information</i></p>
Run OUI with CVU and use fixup scripts	<p>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.</p> <p>You can also run CVU commands manually to check system readiness. For more information, see:</p> <p><i>Oracle Clusterware Administration and Deployment Guide</i></p>
Update and run Oracle ORAchk for runtime and upgrade checks, or runtime health checks	<p>Oracle recommends that you update to the latest version of Oracle ORAchk.</p> <p>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.</p> <p>Use the Oracle ORAchk Upgrade Readiness Assessment to obtain an automated upgrade-specific system health check for upgrades to 11.2.0.3, 11.2.0.4, 12.1.0.1, 12.1.0.2, 12.2, 18c, and 19c. For example:</p> <ul style="list-style-type: none"> • Before you perform a fresh database installation: <pre>./orachk -profile preinstall</pre> • To upgrade your existing database to a higher version or release: <pre>./orachk -u -o pre</pre> <p>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 1268927.1 for more information about Oracle ORAchk support.</p> <p>https://support.oracle.com/rs?type=doc&id=1268927.2</p>
Verify if Oracle Grid Infrastructure is installed	<p>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.</p> <p>For Oracle Real Application Clusters (Oracle RAC) installations, ensure that you have installed and configured Oracle Grid Infrastructure for a cluster.</p>

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

Check	Task
Check running Oracle processes, and shut down if necessary	<ul style="list-style-type: none"> • 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.
Ensure <code>cron</code> jobs do not run during installation	<p>If the installer is running when daily <code>cron</code> jobs start, then you may encounter unexplained installation problems if your <code>cron</code> job is performing cleanup, and temporary files are deleted before the installation is finished. Oracle recommends that you complete installation before daily <code>cron</code> jobs are run, or disable daily</p> <pre data-bbox="672 835 730 856">cron</pre> <p>jobs that perform cleanup until after the installation is completed.</p>
Obtain your My Oracle Support account information.	<p>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:</p> <p>https://support.oracle.com/</p>
Decide Oracle Database management tool	<p>By default, Oracle Database is managed by Oracle Enterprise Manager Database Express.</p> <p>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:</p> <ul style="list-style-type: none"> • OMS host • OMS port • EM admin username • EM admin password • Specify password of ASMSNMP user <p>For more information, see:</p> <ul style="list-style-type: none"> • <i>Oracle Database 2 Day DBA</i> • <i>Oracle Enterprise Manager Cloud Control Administrator's Guide</i>

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

Check	Task
Review memory allocation and Automatic Memory Management feature	<p>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.</p> <p>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.</p> <p>For more information, see: <i>Oracle Database Administrator's Guide</i></p>
Determine superuser (root) privilege delegation option for installation	<p>During a database or grid infrastructure installation, you are asked to run configuration scripts as the <code>root</code> user.</p> <p>You can either run these scripts manually as <code>root</code> when prompted, or you can provide configuration information and passwords using a root privilege delegation option.</p> <p>To run <code>root</code> scripts automatically, select Automatically run configuration scripts during installation.</p> <ul style="list-style-type: none"> • 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 <code>root</code>. 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	<p>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</p>

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	<p>To install Oracle RAC 19c, you must have Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) 19c installed on your cluster. The Oracle Clusterware version must be equal to or greater than the Oracle RAC version that you plan to install.</p> <p>See Also: <i>Oracle Database Upgrade Guide</i> <i>Oracle Grid Infrastructure Installation Guide</i></p>
Ensure that Oracle RAC Databases You are Installing are Compatible with Existing Databases	<p>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 19c when you perform an Oracle Grid Infrastructure 19c installation.</p> <p>If you have an existing Oracle home, then you can create a new Oracle home and install Oracle Database 19c 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.</p>
Migrate files from RAW devices to Supported Storage	<p>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.</p>
Prepare to upgrade all existing nodes	<p>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.</p>
Ensure that Oracle RAC database version is equal to or older than the version of Oracle Clusterware	<p>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:</p> <ul style="list-style-type: none"> • If your servers use Oracle Grid Infrastructure 19c, then you can have an Oracle Database 19c single-instance database running on one node, and separate Oracle Database 11g Release 2 (11.2), Oracle RAC 12c Release 1 (12.1), Oracle RAC 12c Release 2 (12.2), Oracle RAC 18c, or Oracle RAC 19c databases also running on the cluster. • You cannot have Oracle Grid Infrastructure 11g Release 2 (11.2) installed on your cluster, and install Oracle RAC 12c Release 1 (12.1).

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](#)
Before you start the setup wizards for your Oracle Database or Oracle Grid Infrastructure installation, decide if you want to use any of the available image-creation options.
- [Downloading the Software from Oracle Software Delivery Cloud Portal](#)
You can download the software from Oracle Software Delivery Cloud.
- [Deciding Between Multitenant Container Databases and Non-CDBs in Oracle RAC](#)
Review the information to decide how to deploy your Oracle Database using the CDB or Non-CDB options.
- [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](#)
Oracle Fleet Patching and Provisioning (Oracle FPP) provides a standardized approach for provisioning, patching and upgrading software homes on any number of nodes from a single cluster.

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 Release Update Revisions (Revisions).

 **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

Before you start the setup wizards for your Oracle Database or Oracle Grid Infrastructure installation, decide if you want to use any of the available image-creation 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
<code>-createGoldImage</code>	Creates a gold image from the current Oracle home.
<code>-destinationLocation</code>	Specify the complete path, or location, where the gold image will be created.
<code>-exclFiles</code>	Specify the complete paths to the files to be excluded from the newly created gold image.
<code>-help</code>	Displays help for all the available options.

For example:

```
./runInstaller -createGoldImage -destinationLocation /tmp/my_db_images
```

```
./gridSetup.sh -createGoldImage -destinationLocation /tmp/  
my_grid_images
```

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/19.0.0/dbhome_1/relnotes` is the file to be excluded in 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.

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

Review the information to decide how to deploy your Oracle Database using the CDB or Non-CDB options.

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

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

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

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

 **See Also:**

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

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.

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.
 - [Applying Patches During an Oracle Database Installation or Upgrade](#)
Starting with Oracle Database 18c, you can download and apply Release Updates (RUs) during an Oracle Database installation or upgrade.

Installing Oracle RAC and Oracle RAC One Node Database Software

Install Oracle RAC or Oracle RAC One Node software.

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/19.0.0/dbhome_1
$ chgrp oinstall /u01/app/oracle/product/19.0.0/dbhome_1
$ cd /u01/app/oracle/product/19.0.0/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/19.0.0/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**.
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](#)
Oracle Fleet Patching and Provisioning (Oracle FPP) provides a standardized approach for provisioning, patching and upgrading software homes on any number of nodes from a single cluster.
- [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.

Applying Patches During an Oracle Database Installation or Upgrade

Starting with Oracle Database 18c, you can download and apply Release Updates (RUs) during an Oracle Database installation or upgrade.

1. Download the patches you want to apply from My Oracle Support:
<https://support.oracle.com>

2. Select the **Patches and Updates** tab to locate the patch.

Oracle recommends that you select **Recommended Patch Advisor**, and enter the product group, release, and platform for your software.

3. Move the patches to an accessible directory like `/tmp`.
4. Change to the Oracle Database home directory:

```
$ cd $ORACLE_HOME
```

5. Apply Release Updates (RUs) during the installation or upgrade process:

```
$ ./runInstaller -applyRU patch_directory_location
```

6. Complete the remaining steps in the Oracle Database configuration wizard to complete the installation or upgrade.

About Deploying Oracle Databases Using Oracle Fleet Patching and Provisioning

Oracle Fleet Patching and Provisioning (Oracle FPP) provides a standardized approach for provisioning, patching and upgrading software homes on any number of nodes from a single cluster.



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, 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. Oracle Multitenant is also supported.
- 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.
- Ensures easy 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 DBCA](#)
Before using DBCA to create Oracle RAC databases, verify if your system is prepared for configuration changes.
- [Tasks to Complete Before Using DBCA to Create any Oracle RAC Database](#)
Before you can create an Oracle RAC database using DBCA you must configure your system to meet the software requirements, if you did not complete your system configuration during the Oracle Grid Infrastructure installation.
- [Selecting DBCA Options to Create an Oracle RAC or Oracle RAC One Node Database](#)
Review this information to use Database Configuration Assistant (DBCA) to create Oracle RAC or Oracle RAC One Node Database.
- [Using DBCA to Create an Oracle RAC One Node Database](#)
If you have selected to install only the Oracle RAC software on cluster nodes, then you can use Oracle Database Configuration Assistant (DBCA) to configure Oracle RAC One Node.
- [Installing the Oracle Database Vault Option](#)
Installing and configuring Oracle Database Vault requires actions during and after installation.
- [Deleting an Oracle RAC Database Using DBCA](#)
Deleting an Oracle RAC database using DBCA involves first deleting the database, and then removing the database's initialization parameter files, instances, Optimal Flexible Architecture (OFA) structure, and the Oracle network configuration for the database.

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 both policy-managed and administrator-managed databases. With Oracle DBCA, you can also create site-specific tablespaces as part of database creation. If you have data file requirements that differ from those offered by 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 Database Services After Installation](#)
Use the Server Control Utility (SRVCTL), or use Oracle Enterprise Manager Database Express, or Oracle Enterprise Manager Cloud Control for all administration and monitoring of database services for an Oracle RAC database.

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 administrator-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.

Starting with Oracle Database 12c Release 2 (12.2), 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, database operations on character data may be slightly slower when compared to single-byte database character sets, such as WE8ISO8859P1 or WE8MSWIN1252. Storage space requirements for text in most languages that use characters outside of the ASCII repertoire are higher in AL32UTF8 compared to legacy character sets supporting the language. English data may require more space only if stored in CLOB (character large object) columns. Storage for non-character data types, such as NUMBER or DATE, does not depend on a character set. The universality and flexibility of Unicode usually outweighs these additional costs.

Consider legacy character sets only when the database needs to support a single group of languages and the use of a legacy character set is critical for fulfilling compatibility, storage, or performance requirements. The database character set to be selected in this case is the character set of most clients connecting to this database.

The database character set of a multitenant container database (CDB) determines which databases can be plugged in later. Ensure that the character set you choose for the CDB is compatible with the database character sets of the databases to be plugged into this CDB. If you use Unicode AL32UTF8 as your CDB character set, then you can plug in a pluggable database (PDB) in any database character set supported by Oracle Database (with the exception of EBCDIC-based character sets).

 **See Also:**

Oracle Database Globalization Support Guide for more information about choosing a database character set for a multitenant container database (CDB)

Managing Database Services After Installation

Use the Server Control Utility (SRVCTL), or use Oracle Enterprise Manager Database Express, or Oracle Enterprise Manager Cloud Control for all administration and monitoring of database services for an Oracle RAC database.

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

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 11g Release 2 (11.2), 12c Release 1 (12.1), 12c Release 2 (12.2), or 18c installation, and you install Oracle Grid Infrastructure 19c either to coexist with or to upgrade the Oracle Grid Infrastructure 11.2, 12.1, 12.2, or 18c installation, then most installation types automatically migrate the existing listener to the 19c 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 DBCA

Before using DBCA to create Oracle RAC databases, verify if your system is prepared for configuration changes.

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

```
/u01/app/19.0.0/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.

For example, to verify that your system is prepared for an Oracle Database with Oracle RAC installation on a two-node cluster with nodes `node1` and `node2`, with the Oracle Grid Infrastructure home path `/u01/app/19.0.0/grid`, and with the Oracle home path `/u01/app/oracle/product/19.0.0/dbhome_1`, enter the following command:

```
$ /u01/app/19.0.0/grid/bin/cluvfy stage -pre dbcfg -fixup -n node1,node2 -d\  
/u01/app/oracle/product/19.0.0/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 DBCA to Create any Oracle RAC Database

Before you can create an Oracle RAC database using DBCA you must configure your system to meet the software requirements, if you did not complete your system configuration during the Oracle Grid Infrastructure installation.

- [Load SSH Keys Into Memory Before Starting DBCA](#)
In an Oracle RAC environment, you must load SSH keys into memory for the terminal session where you start DBCA. If you do not do this, then you receive user equivalency errors when you attempt to start DBCA.
- [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](#)
Before starting DBCA to configure an Oracle RAC database, you must have installed Oracle Grid Infrastructure for a cluster, and you must have configured shared storage areas for Oracle RAC files.

Load SSH Keys Into Memory Before Starting DBCA

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

If you use a pass phrase on your system for SSH, then you must provide the pass phrase to load the SSH keys.

Use the following commands to load SSH keys:

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

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

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

Before starting DBCA to configure an Oracle RAC database, you must have installed Oracle Grid Infrastructure for a cluster, and you must have configured shared storage areas for Oracle RAC files.

Storage administration tasks require the `SYSASM` system privileges, which are granted to members of the `OSASM` operating system group. This group may not be the same as the `OSDBA` group, whose members are granted the `SYSDBA` system privileges.

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

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

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



Note:

You can no longer set up email notification for Oracle RAC databases either from DBCA or Oracle Universal Installer (OUI).

- [Starting DBCA](#)
To start DBCA, connect as the installation owner account (for example, `oracle`) to one of your nodes where Oracle RAC is installed, load SSH keys into memory, and enter the command `dbca` from the `$ORACLE_HOME/bin` directory.
- [Cluster Detection and Node Selection when Using DBCA](#)
When you start DBCA, it automatically shows Oracle RAC options if it detects from the central Oracle Inventory that the Oracle home is enabled for Oracle RAC.
- [Using DBCA to Select Storage to Use With any Oracle RAC Database](#)
You can choose to use either Oracle ASM disk groups or a supported cluster file system as storage.
- [Selecting Server Pool Option with DBCA](#)
When creating a database using Database Configuration Assistant (DBCA), you can choose the server pools to host policy-managed databases.
- [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 DBCA

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



Note:

You no longer need to set the operating system environment variables `ORACLE_HOME` to the Oracle RAC database home, or `ORACLE_UNQNAME` to the database unique name.

Cluster Detection and Node Selection when Using DBCA

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

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

```
/u01/app/19.0.0/grid/bin/cluvfy/cluvfy.sh stage -post crsinst -n nodelist
```

For example, with the mountpoint `/u01/app/19.0.0/grid`, and nodes `node1` and `node2`, run the following command:

```
$ /u01/app/19.0.0/grid/bin/cluvfy stage -post crsinst -n node1,node2
```

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

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

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

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

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

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

Selecting Server Pool Option with DBCA

When creating a database using Database Configuration Assistant (DBCA), you can choose the server pools to host policy-managed databases.

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

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

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

After installation of Oracle Real Application Clusters (Oracle RAC) software, start Oracle Database Configuration Assistant (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 DBCA](#)
You can configure Oracle Database Vault after installation using Oracle Database Configuration Assistant (DBCA), 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 DBCA

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

To install Oracle Database Vault using DBCA:

1. After installing Oracle RAC, create the database.
2. Start DBCA and select the option **Configure Database**.
3. In the component list, select **Oracle Label Security** and **Oracle Database Vault**.
4. Provide the required Oracle Database Vault user IDs and passwords to proceed with configuration. To enable a separate Oracle Database Vault administrator, choose to configure the DV_ACCTMGR user.
5. After you have finished, restart each database instance to finish the software configuration.

 **See Also:**

Oracle Database Vault Administrator's Guide for information about using Oracle Data Guard with Oracle Database Vault

Deleting an Oracle RAC Database Using DBCA

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

To delete a database using DBCA:

1. Start DBCA on one of the nodes:
 - Run the `dbca` command from the `$ORACLE_HOME/bin` directory.DBCA displays the Operations page, displaying different database deployment options.
2. Select **Delete a database**, and click **Next**. DBCA displays a list of all Oracle RAC and single-instance databases running from the Oracle home where DBCA is run.
3. If your user ID and password are not operating-system authenticated, then the List of Cluster Databases page displays the user name and password fields. If these fields appear, then enter a user ID and password for a user account that has `SYSDBA` privileges.
4. Select the database to delete, and click **Finish**.

After you click **Finish**, DBCA displays a dialog box to confirm the database and instances that you have configured DBCA to delete.
5. Click **OK** to begin the deletion of the database and its associated files, services, and environment settings, or click **Cancel** to stop the operation.

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

At this point, you have accomplished the following:

- Deleted the selected Oracle RAC database from the cluster
- Deleted high availability services assigned to the Oracle RAC database
- Deleted the Oracle Net configuration for the Oracle RAC database
- Deconfigured Oracle Enterprise Manager for the Oracle RAC database
- Deleted the OFA directory file structure for that Oracle RAC database from the cluster
- Deleted the Oracle RAC database data files

4

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 Release Update Revisions (RUR) 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](#)
On each node, in the installation owner user profile file, set the environment variables `ORACLE_BASE` and `ORACLE_HOME`, and `ORACLE_SID`; also add `ORACLE_HOME/bin` to the path environment variable.
- [Recompiling All Invalid Objects](#)
Identify and recompile invalid objects on the CDB and PDBs using the `catcon` utility to run `utlrp.sql` after you install, patch, or upgrade a database.

- [Configuring Services on Oracle RAC and Oracle RAC One Node CDBs](#)
During installation, if you select a multitenant container database (CDB), and configure pluggable databases (PDBs), then you must add services to the PDBs after installation.
- [Copying Oracle ASM Password File For Oracle RAC One Node Database](#)
After installing Oracle RAC One Node database, copy the Oracle ASM password file, if configured, to all candidate nodes configured for fail over, unless you use shared password files stored in Oracle ASM.

Downloading Release Update Patches

Download and install Release Updates (RU) and Release Update Revisions (RUR) patches for your Oracle software after you complete installation.

Starting with Oracle Database 18c, Oracle provides quarterly updates in the form of Release Updates (RU) and Release Update Revisions (RUR). 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. Log in to My Oracle Support website.

 **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

On each node, in the installation owner user profile file, set the environment variables `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/19.0.0/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.

Recompiling All Invalid Objects

Identify and recompile invalid objects on the CDB and PDBs using the `catcon` utility to run `utlrp.sql` after you install, patch, or upgrade a database.

Note:

If you upgraded using the AutoUpgrade utility, then AutoUpgrade automatically takes care of this task during the upgrade. You do not need to perform this task.

Oracle recommends that you use the `catcon.pl` utility to run `utlrbp.sql` on all containers in your container database (CDB). The `utlrbp.sql` script recompiles all invalid objects. Run the script immediately after installation, to ensure that users do not encounter 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 `utlrbp.sql`. For example:

```
$ORACLE_HOME/perl/bin/perl catcon.pl --n 1 --e --b utlrbp --d  
''. '' utlrbp.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 `utlrbp`.

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 `utlrbp.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.

For more information about `catcon` utility syntax and options, refer to *Oracle Multitenant Administrator's Guide*.

Related Topics

- Syntax and Parameters for `catcon.pl`

Configuring Services on Oracle RAC and Oracle RAC One Node CDBs

During installation, if you select a multitenant container database (CDB), and configure pluggable databases (PDBs), then you must add services to the PDBs after installation.

If you do not add services to PDBs, and then the Oracle RAC One Node CDB fails over to another node, or you manually relocate the CDB to another node, then by default, all PDBs associated with the CDB that do not have registered services are restarted in `MOUNTED` state.

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

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

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

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

See Also:

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

Copying Oracle ASM Password File For Oracle RAC One Node Database

After installing Oracle RAC One Node database, copy the Oracle ASM password file, if configured, to all candidate nodes configured for fail over, unless you use shared password files stored in Oracle ASM.

If you install Oracle RAC One Node database using Database Configuration Assistant (DBCA), 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](#)
Install the latest version of Oracle Autonomous Health Framework to perform proactive health checks and collect diagnostics data for the Oracle software stack.

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

Install the latest version of Oracle Autonomous Health Framework to perform proactive health checks and collect diagnostics data for the Oracle software stack.

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:

- Standalone Oracle Database
- Oracle Grid Infrastructure
- Oracle Real Application Clusters
- Maximum Availability Architecture (MAA) Validation
- Upgrade Readiness Validations
- Oracle GoldenGate

Oracle Autonomous Health Framework is pre-installed with Oracle Database. However, Oracle recommends that you update to the latest version of Oracle Autonomous Health Framework 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>

Product-Specific Postinstallation Tasks

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

- [Configuring Oracle Database Vault](#)
OUI installs Oracle Database Vault by default when you install the Oracle RAC software, but you must register Oracle Database Vault with the Oracle RAC database and create database user accounts before you can use it.
- [Configuring Oracle Database Security Settings](#)
Use Oracle Database Configuration Assistant (DBCA) commands to change security settings after installation.
- [Configuring Oracle Label Security](#)
After installation, you must configure Oracle Label Security in a database before you use it.
- [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](#)
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.

Configuring Oracle Database Vault

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

You must create the Database Vault Owner user and, optionally, the Database Vault Account Manager administrative user accounts.

Configuring Oracle Database Security Settings

Use Oracle Database Configuration Assistant (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 (DBCA) options. By design, the DBCA graphical user interface (GUI) does not have the option to enable or disable secure configuration. For example, to enable the security settings after installation, use a command of the following form, where `myRACdb1.example.com` is either the name of the local database instance, or the value set for the initialization parameter `DB_UNIQUE_NAME`:

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

Configuring Oracle Label Security

After installation, you must configure Oracle Label Security in a database before you use it.

You can configure Oracle Label Security in two ways: with Oracle Internet Directory integration, 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

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 ASM Cluster File System (Oracle ACFS), or a supported network file system (NFS) using the Direct NFS Client.

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

Note:

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

See Also:

Oracle Grid Infrastructure Installation Guide for more information about configuring storage

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
<code>olap</code>	Oracle OLAP
<code>partitioning</code>	Oracle Partitioning
<code>rat</code>	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 Capacity Management](#)
Oracle Clusterware uses policy-based management of servers and resources used by Oracle databases or 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 a server pool with DBCA while creating an Oracle RAC database, but Oracle recommends that you create server pools before you deploy database software and databases.
- [Oracle RAC One Node and Server Pools](#)
Review the following information about Oracle RAC One Node and server pools.

Policy-Managed Clusters and Capacity Management

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

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

- [Server Pools and Server Categorization](#)
You can manage servers dynamically using server pools by identifying servers distinguished by particular attributes, a process called **server categorization**. In this way, you can manage clusters made up of heterogeneous nodes.
- [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 Server Pools](#)
When Oracle Clusterware is installed, two server pools are created automatically: *Generic* and *Free*.



See Also:

Oracle Real Application Clusters Administration and Deployment Guide

Server Pools and Server Categorization

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

Server Pools and Policy-Based Management

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

Policy-based management:

- Enables dynamic capacity assignment when needed to provide server capacity in accordance with the priorities you set with policies
- Enables allocation of resources by importance, so that applications obtain the required minimum resources, whenever possible, and so that lower priority applications do not take resources from more important applications
- Ensures isolation where necessary, so that you can provide dedicated servers in a cluster for applications and databases
- Enables policies to be configured to change pools in accordance with business needs or application demand, so that pools provide the right service at the right time

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

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



See Also:

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

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

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 Generic Server Pool](#)
The Generic server pool stores any Oracle Database that is not policy managed.

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

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.

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

Oracle also recommends that you:

- Enable role separation before you create the first server pool in the cluster.
- Create and manage server pools using configuration policies and a respective policy set.

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

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

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

 **See Also:**

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

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 Net Configuration Assistant (NETCA) and Database Configuration Assistant (DBCA) configure your environment to meet the requirements for database creation and Oracle Enterprise Manager discovery of Oracle RAC databases.
- [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](#)
When you create the database, Oracle Database creates an SPFILE in the file location that you specify. This location can be either an Oracle ASM disk group or a cluster file system.
- [About ORATAB Configuration for Oracle RAC](#)
Oracle creates an entry for each Oracle RAC database in the `oratab` configuration file.
- [Database Components Created Using Database Configuration Assistant](#)
Understand the database components that 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](#)
Users 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](#)
If you use a naming method other than Easy Connect, then additional configuration of Oracle Net Services may be required.

Understanding the Configured Environment in Oracle RAC

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



Note:

Configuration files are created on each node in your cluster database.

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

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 <code>oraInventory</code> directory for each server, and other privileges granted to Oracle binary installation owner users.
OSDBA	dba	<code>SYSDBA</code> system privileges for an Oracle Database, which includes all system privileges for the database.
OSOPER	oper	<code>SYSOPER</code> startup and shutdown system privileges for an Oracle Database.
OSBACKUPDBA	backupdba	<code>SYSBACKUP</code> backup and recovery system privileges for an Oracle Database.
OSDGDBA	dgdba	<code>SYSDG</code> system privileges to administer and monitor Oracle Data Guard.
OSKMDBA	kmdba	<code>SYSKM</code> system privileges for encryption key management for applications such as Oracle Wallet Manager.
OSASM	asmadmin	<code>SYSASM</code> system privileges for Oracle ASM on a cluster, which includes all system privileges for Oracle ASM storage.
OSOPER for ASM	asmoper	<code>SYSOPER</code> startup and shutdown system privileges for Oracle ASM on the cluster.
OSDBA for ASM	asmdba	<code>SYSDBA</code> 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	<code>SYSRAC</code> 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

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

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



See Also:

Oracle Real Application Clusters Administration and Deployment Guide for information about creating and modifying SPFILE

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 release is Oracle Database 12c Release 1 (12.1) or earlier
- 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 Database Configuration Assistant

Understand the database components that 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 <code>DRSYS</code> (contains data for Oracle Text), <code>CWMLITE</code> (contains the OLAP schemas), <code>XDB</code> (for XML features), <code>ODM</code> (for Oracle Data Mining), <code>INDEX</code> and <code>EXAMPLE</code> schemas.

Table 6-2 (Cont.) Tablespace Names Used with Oracle Real Application Clusters Databases

Tablespace Name	Contents
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 <code>ANALYZE COMPUTE STATISTICS</code> on a very large table, or the constructs <code>GROUP BY</code> , <code>ORDER BY</code> , or <code>DISTINCT</code> .
UNDOTBS _n	Contains undo tablespaces for each instance that DBCA creates for automatic undo management.
USERS	Consists of application data. As you create and enter data into tables, Oracle Database fills this space with your data.

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

Users 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. A service is identified by a service name, such as `sales.example.com`. A client uses a service name to identify the database it must access.
- [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 19c 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. A service is identified by a service name, such as `sales.example.com`. A client uses a service name to identify the database it must access.

During installation, Oracle RAC databases are configured with a default database service that has the same name as the database. This service can be used for performing database management tasks. Create additional services for client and application connections to the database.

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

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 simple TCP/IP address, which consists of a host name, and an optional port and service name. If you use this method, then no naming or directory system is required.

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, rather than node VIPs used in releases before Oracle Grid Infrastructure 11g Release 2 (11.2). Clients connecting to Oracle RAC databases using SCANS do not have to be configured with addresses of each node that hosts a particular

database or database instance. For example, if you configure policy-managed server pools for a cluster, then connecting to the database using a SCAN enables connections to server pools in that database, regardless of which nodes are allocated to the server pool. You can add or remove nodes from the database without having to reconfigure clients connecting to the database.

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 19c 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 19c 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
```

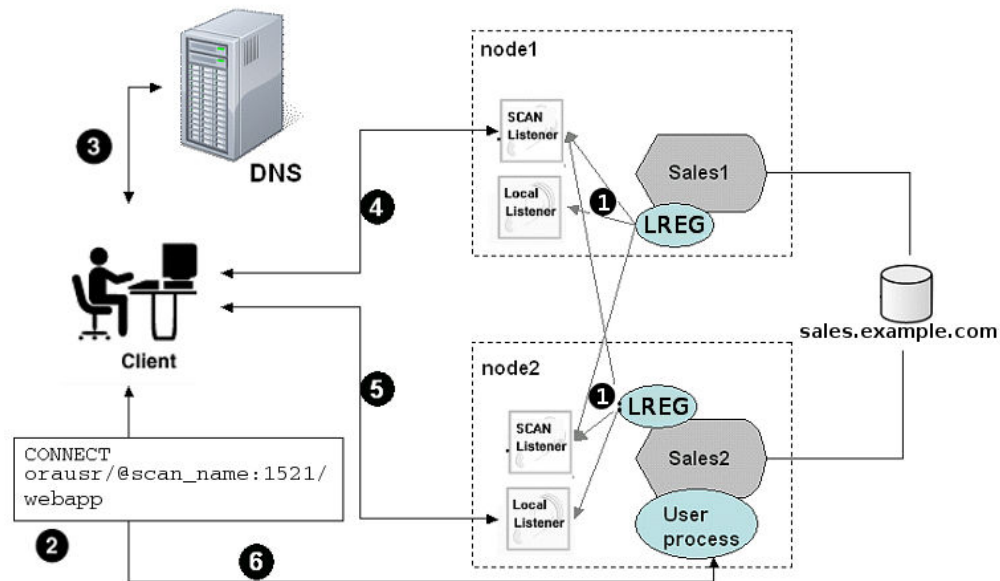
 **Note:**

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. By default, DBCA configures your Oracle RAC database with dedicated servers, not shared servers.

Load Balancing of Connections to Oracle RAC Databases

Services coordinate their sessions by registering their **workload**, or the amount of work they are currently handling, with the local listener and the SCAN listeners.

Clients are redirected by the SCAN listener to a local listener on the least-loaded node that is running the instance for a particular service. This feature is called **load balancing**. The local listener either directs the client to a dispatcher process (if you configured the database to use shared servers), or directs the client to a dedicated server process.

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

See Also:

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

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 surviving node, enabling fast notification of the failure to the clients connecting through the VIP. If the application and client are configured with transparent application failover options, then the client is reconnected to the surviving node.

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

If you use a naming method other than Easy Connect, then additional configuration of Oracle Net Services may be 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.

Review these topics for information about the Oracle Net Services configuration files and parameters for an Oracle RAC database:

- [Database Initialization Parameters for Database Service Registration](#)
An Oracle Database 19c 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 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.
- [Net Service Names Created by DBCA](#)
DBCA creates net service names for connections.
- [Listener Configuration and the `listener.ora` File](#)
In Oracle RAC environments, Oracle recommends that you let the Oracle Agent manage Oracle listeners for Oracle Databases.
- [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 19c 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 installation process creates a `tnsnames.ora` file on each node. This file acts as a repository of net service names. Each net service name is associated with a connect identifier. A **connect identifier** is an identifier that maps a user-defined name to a connect descriptor.

A connect descriptor contains the following information:

- The network route to the service, including the location of the listener through a protocol address
- The `SERVICE_NAME` parameter, with the value set to the name of a database service

 **Note:**

The `SERVICE_NAME` parameter that you use in the `tnsnames.ora` file is singular, because you can specify only one service name. The `SERVICE_NAME` parameter is not the same as the `service_names` database initialization parameter. The `service_names` database parameter defaults to the global database name, a name comprising the `db_name` and `db_domain` parameters in the initialization parameter file. When you add service names using SRVCTL or Oracle Enterprise Manager Cloud Control, it lists additional cluster-managed services for the database.

The `tnsnames.ora` file is located in both the `Grid_home/network/admin` and `Oracle_home/network/admin` directories. By default, the `tnsnames.ora` file is read from the Grid home when Oracle Grid Infrastructure is installed.

With Oracle Clusterware 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=
(AADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=192.168.0.61) (PORT=1521))
(AADDRESS=(PROTOCOL=TCP) (HOST=192.168.0.61) (PORT=2012)))'
scope=BOTH SID='OCRL1';
```



See Also:

- *Oracle Database Administrator's Guide*
- *Oracle Database Net Services Administrator's Guide* for more information about the `tnsnames.ora` file

Net Service Names Created by DBCA

DBCA creates net service names for 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.

If you use DBCA to create an Oracle RAC database that is a multitenant container database (CDB), then DBCA creates a database service that has the same name as the database. 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 CDB. However, if you use DBCA to add a pluggable database (PDB) to an existing CDB, then DBCA does not create a database service for the new PDB.

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

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

**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. The `listener.ora` file is located in the `Grid_home/network/admin` directory.
- [Remote Listeners for an Oracle RAC Database](#)
A **remote listener** is a listener residing on one computer that redirects connections to a database instance on another computer. For example, SCAN listeners are remote listeners.
- [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. It can include the protocol addresses it is accepting connection requests on, a list of the database and other services it is listening for, and control parameters used by the listener.

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 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.

 **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** is a listener residing on one computer that redirects connections to a database instance on another computer. For example, SCAN listeners are remote 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.

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 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.

 **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 19c 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 19c.

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. It can include the protocol addresses it is accepting connection requests on, a list of the database and other services it is listening for, and control parameters used by the listener.

You can modify the configuration of the listeners used by Oracle Clusterware and Oracle RAC with Server Control Utility (`SRVCTL`) commands, or by using `NETCA`. Manual editing of the `listener.ora` file is not required.

Each listener is configured with one or more protocol addresses that specify its listening endpoints. The listener agent dynamically updates endpoints with the listener. Starting with Oracle Database 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/19.0.0/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

These topics describe how to remove Oracle software and configuration files.

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 standalone 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)
- 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 19c Oracle home to remove Oracle software from an existing 11.2.0.4 Oracle home.
- Starting with Oracle Database 12c Release 1 (12.1.0.2), the `roothas.sh` script replaces the `roothas.pl` script in the Oracle Grid Infrastructure home for Oracle Restart, and the `rootcrs.sh` script replaces the `rootcrs.pl` script in the Grid home for Oracle Grid Infrastructure for a cluster.

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 Automatic Storage Management 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.

To identify all instances associated with the Oracle home you want to remove, enter the following command, where `dbname` is the name of the database:

```
$ srvctl status database -d dbname
```

Alternately, you can check for registered instances by viewing the `oratab` file:

AIX, HP-UX, or Linux:

```
$ more /etc/oratab
```

Oracle Solaris:

```
$ more /var/opt/oracle/oratab
```

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

```
+ASM1:/u01/app/19.0.0/grid:N  
CUST:/u01/app/oracle/product/19.0.0/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/19.0.0/grid`) and the `CUST` Oracle database instance are associated with the Oracle home directory `/u01/app/oracle/product/19.0.0/dbhome_1`.

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
<code>-silent</code>	<p>Use this flag to run <code>deinstall</code> in noninteractive mode. This option requires one of the following:</p> <ul style="list-style-type: none"> • A working system that it can access to determine the installation and configuration information. The <code>-silent</code> 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. <p>You can generate a response file to use or modify by running <code>deinstall</code> with the <code>-checkonly</code> flag. <code>deinstall</code> then discovers information from the Oracle home to deinstall and deconfigure. It generates the response file that you can then use with the <code>-silent</code> option.</p> <p>You can also modify the template file <code>deinstall.rsp.tmpl</code>, located in the <code>\$ORACLE_HOME/deinstall/response</code> directory.</p>
<code>-checkonly</code>	<p>Use this flag to check the status of the Oracle software home configuration. Running <code>deinstall</code> with the <code>-checkonly</code> flag does not remove the Oracle configuration. The <code>-checkonly</code> flag generates a response file that you can then use with the <code>deinstall</code> command and <code>-silent</code> option.</p>
<code>-paramfile</code> <i>complete path of input response file</i>	<p>Use this flag to run <code>deinstall</code> 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.</p> <p>The default location of the response file is <code>\$ORACLE_HOME/deinstall/response</code>.</p>

Parameter	Description
<code>-params [name1=value name2=value name3=value ...]</code>	Use this flag with a response file to override one or more values to change in a response file you have created.
<code>-o complete path of directory for saving response files</code>	Use this flag to provide a path other than the default location where the response file (<code>deinstall.rsp.tmpl</code>) is saved. The default location of the response file is <code>\$ORACLE_HOME/deinstall/response</code> .
<code>-tmpdir complete path of temporary directory to use</code>	Use this flag to specify a non-default location where <code>deinstall</code> writes the temporary files for the deinstallation.
<code>-logdir complete path of log directory to use</code>	Use this flag to specify a non-default location where <code>deinstall</code> writes the log files for the deinstallation.
<code>-local</code>	Use this flag on a multinode environment to deinstall Oracle software in a cluster. When you run <code>deinstall</code> with this flag, it deconfigures and deinstalls the Oracle software on the local node (the node where <code>deinstall</code> is run). On remote nodes, it deconfigures Oracle software, but does not deinstall the Oracle software.
<code>-skipLocalHomeDeletion</code>	Use this flag in Oracle Grid Infrastructure installations on a multinode environment to deconfigure a local Grid home without deleting the Grid home.
<code>-skipRemoteHomeDeletion</code>	Use this flag in Oracle Grid Infrastructure installations on a multinode environment to deconfigure a remote Grid home without deleting the Grid home.
<code>-help</code>	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/19.0.0/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/19.0.0/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/19.0.0/grid`

```
$ cd /u01/app/19.0.0/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.

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/19.0.0/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/19.0.0/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/19.0.0/dbhome_1/  
SPFILE_LOC.sidb=
```

A

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](#)
Review this topic 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](#)
You can perform a noninteractive, or *silent* configuration of Oracle RAC using DBCA. To perform a silent configuration, you must have completed an Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) installation, run the `root.sh` script from the Oracle Database home, and defined the Oracle home directory environment variable.
- [Using DBCA Commands for Noninteractive \(Silent\) Configuration for Oracle RAC](#)
Review this topic for the command syntax to create an Oracle RAC database using 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.
- [Postinstallation Configuration Using the ConfigToolAllCommands Script](#)
You can create and run a response file configuration after installing Oracle software. The `configToolAllCommands` script requires users to create a second response file, of a different format than the one used for installing the product.

- [Running Configuration Assistants Using Response Files](#)
You can run configuration assistants in response file or silent mode to configure and start Oracle software after it is installed on the system. To run configuration assistants in response file or silent mode, you must copy and edit a response file template.

Using DBCA to Generate Installation Scripts for Oracle RAC

Review this topic to generate scripts to create an Oracle RAC database, create a database using the generated scripts, and prepare the database for use.

Complete the following steps:

1. Start DBCA and select your preferred options to build the Oracle RAC database.
On the Creation Options page of your DBCA session, deselect Create Database and select **Generate Database Creation Scripts** before you click **Finish**. You can accept the default destination directory for the scripts, or browse for a different location. In either case, 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 Net Configuration Assistant (NETCA) and Database Configuration Assistant (DBCA) configure your environment to meet the requirements for database creation and Oracle Enterprise Manager discovery of Oracle RAC databases.

About DBCA Noninteractive (Silent) Configuration for Oracle RAC

You can perform a noninteractive, or *silent* configuration of Oracle RAC using DBCA. To perform a silent configuration, you must have completed an Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) installation, run the `root.sh` script from the Oracle Database home, and defined the Oracle home directory environment variable.

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

Oracle provides templates for the following two workload types:

- General purpose or transaction processing
- Data warehouse

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

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

Review this topic for the command syntax to create an Oracle RAC database using DBCA.

You can use the following command syntax to create an Oracle RAC database using the general purpose template, placing the data files in an existing Oracle ASM disk group. Nodes `node1` and `node2` are the cluster nodes on which Oracle RAC database instances are created. The disk group name is `+ASMgrp1`, and `password` is a placeholder for a password. The passwords can be all the same password or different passwords each time:

```
# su oracle -c "$ORACLE_HOME/bin/dbca -silent -createDatabase -templateName
General_Purpose.dbc -gdbName $DBNAME -sid $ORACLE_SID -sysPassword password
-systemPassword password -sysmanPassword password -dbsnmpPassword password
-emConfiguration LOCAL -storageType ASM -diskGroupName ASMgrp1
-datafileJarLocation $ORACLE_HOME/assistants/dbca/templates -nodeinfo
node1,node2 -characterSet WE8ISO8859P1 -obfuscatedPasswords false -
sampleSchema
false -asmSysPassword password"
```

See Also:

Oracle Database Administrator's Guide for a complete description of DBCA commands and options

How Response Files Work

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

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/19.0.0/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.

- [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](#)

Review this information to use response files.

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	<p>Use silent mode for the following installations:</p> <ul style="list-style-type: none">• Complete an unattended installation, which you schedule using operating system utilities such as <code>at</code>.• 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. <p>The installer displays progress information on the terminal that you used to start it, but it does not display any of the installer screens.</p>
Response file	<p>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.</p>

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

Review this information to use response files.

Use the following general steps to install and configure Oracle products using the installer in silent or response file mode:

Note:

You must complete all required preinstallation tasks on a system before running the installer in silent or response file mode.

1. 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. This method is useful for Advanced or software-only 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 A-1 Response Files for Oracle Database and Oracle Grid Infrastructure

Response File	Description
<code>db_install.rsp</code>	Silent installation of Oracle Database.
<code>dbca.rsp</code>	Silent creation and configuration of Oracle Database using Oracle DBCA.
<code>netca.rsp</code>	Silent configuration of Oracle Net using Oracle NETCA.

Table A-1 (Cont.) Response Files for Oracle Database and Oracle Grid Infrastructure

Response File	Description
<code>gridsetup.rsp</code>	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. This method is useful for Advanced or software-only installations.

You can save all the installation steps into a response file during installation by clicking **Save Response File** on the Summary page. You can use the generated response file for a silent installation later.

When you record the response file, you can either complete the installation, or you can exit from the installer on the Summary page, before OUI starts to 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.

To record a response file:

1. Complete preinstallation tasks as for a standard installation.
When you run the installer to record a response file, it checks the system to verify that it meets the requirements to install the software. For this reason, Oracle recommends that you complete all of the required preinstallation tasks and record the response file while completing an installation.
2. Ensure that the Oracle software owner user (typically `oracle`) has permissions to create or write to the Oracle home path that you 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/19.0.0/grid/gridSetup.sh -help
```

The help information appears in a window after some time.

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/19.0.0/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](#)
Starting with Oracle Database 12c release 2 (12.2), 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

Starting with Oracle Database 12c release 2 (12.2), 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 A-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 A-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 A-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, 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.
- If you create a multitenant container database (CDB) with one or more pluggable databases (PDBs), then you must specify a password for the `PDBADMIN` 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`. For example:

For Oracle Database (`oracle` user)

```
oracle.install.asm.SYSASMPassword=password
oracle.install.config.emAdminPassword=password
```

For Oracle Grid Infrastructure (`grid` user)

```
grid.install.asm.SYSASMPassword=password
grid.install.config.emAdminPassword=password
```

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/19.0.0/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/inventory/response/`:

```
$ ./runInstaller -executeConfigTools -responseFile $ORACLE_HOME/inventory/  
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/19.0.0/grid/  
response/grid_2016-01-09_01-03-36PM.rsp [-silent]
```

For Oracle Database:

```
$ ./runInstaller -executeConfigTools -responseFile /u01/app/oracle/  
product/19.0.0/dbhome_1/inventory/response/db_2016-01-09_01-03-36PM.rsp [-  
silent]
```

Postinstallation Configuration Using the ConfigToolAllCommands Script

You can create and run a response file configuration after installing Oracle software. The `configToolAllCommands` script requires users to create a second response file, of a different format than the one used for installing the product.

Starting with Oracle Database 12c Release 2 (12.2), the `configToolAllCommands` script is deprecated and may be desupported in a future release.

- [About the Postinstallation Configuration File](#)
When you run a silent or response file installation, you provide information about your servers in a response file that you otherwise provide manually during a graphical user interface installation.
- [Creating a Password Response File](#)
You can create a password response file and use it with configuration assistants to perform silent installation.

- [Running Postinstallation Configuration Using a Password Response File](#)
Complete this procedure to run configuration assistants with the `configToolAllCommands` script.

Related Topics

- [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.

About the Postinstallation Configuration File

When you run a silent or response file installation, you provide information about your servers in a response file that you otherwise provide manually during a graphical user interface installation.

However, the response file does not contain passwords for user accounts that configuration assistants require after software installation is complete. The configuration assistants are started with a script called `configToolAllCommands`. You can run this script in response file mode by using a password response file. The script uses the passwords to run the configuration tools in succession to complete configuration.

If you keep the password file to use for clone installations, then Oracle strongly recommends that you store the password file in a secure location. In addition, if you have to stop an installation to fix an error, then you can run the configuration assistants using `configToolAllCommands` and a password response file.

The `configToolAllCommands` password response file has the following syntax options:

- `oracle.crs` for Oracle Grid Infrastructure components or `oracle.server` for Oracle Database components that the configuration assistants configure
- `variable_name` is the name of the configuration file variable
- `value` is the desired value to use for configuration.

The command syntax is as follows:

```
internal_component_name|variable_name=value
```

For example:

```
oracle.crs|S_ASMPASSWORD=PassWord
```

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

- If the database is using Oracle Automatic Storage Management (Oracle ASM) for storage, then you must specify a password for the `S_ASMSNMPPASSWORD` variable. If you are not using Oracle ASM, then leave the value for this password variable blank.
- If you create a multitenant container database (CDB) with one or more pluggable databases (PDBs), then you must specify a password for the `S_PDBADMINPASSWORD` variable. If you are not using Oracle ASM, then leave the value for this password variable blank.

Oracle strongly recommends that you maintain security with a password response file:

- Permissions on the response file should be set to 600.
- The owner of the response file should be the installation owner user, with the group set to the central inventory (`oraInventory`) group.

Creating a Password Response File

You can create a password response file and use it with configuration assistants to perform silent installation.

Perform the following steps to create a password response file:

1. Create a response file that has a name of the format *filename.properties*, for example:

```
$ touch pwdrsp.properties
```

2. Open the file with a text editor, and cut and paste the sample password file contents, as shown in the examples, modifying as needed.
3. Change permissions to secure the password response file. For example:

```
$ ls -al pwdrsp.properties
-rw----- 1 oracle oinstall 0 Apr 30 17:30 pwdrsp.properties
```

Example A-4 Password response file for Oracle Grid Infrastructure (grid user)

```
grid.crs|S_ASMPASSWORD=password
grid.crs|S_OMSPASSWORD=password
grid.crs|S_BMCPASSWORD=password
grid.crs|S_ASMMONITORPASSWORD=password
```

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

Example A-5 Password response file for Oracle Grid Infrastructure for a Standalone Server (oracle user)

```
oracle.crs|S_ASMPASSWORD=password
oracle.crs|S_OMSPASSWORD=password
oracle.crs|S_ASMMONITORPASSWORD=password
```

Example A-6 Password response file for Oracle Database (oracle user)

This example provides a template for a password response file to use with the database configuration assistants.

```
oracle.server|S_SYSPASSWORD=password
oracle.server|S_SYSTEMPASSWORD=password
oracle.server|S_EMADMINPASSWORD=password
oracle.server|S_DBSNMPPASSWORD=password
oracle.server|S_ASMSNMPPASSWORD=password
oracle.server|S_PDBADMINPASSWORD=password
```

If you do not want to enable Oracle Enterprise Manager for management, then leave those password fields blank.

Running Postinstallation Configuration Using a Password Response File

Complete this procedure to run configuration assistants with the `configToolAllCommands` script.

1. Create a password response file as described in *Creating a Password File*.
2. Change directory to `$ORACLE_HOME/cfgtoollogs`.
3. Run the configuration script using the following syntax:

```
configToolAllCommands RESPONSE_FILE=/path/name.properties
```

For example:

```
$ ./configToolAllCommands RESPONSE_FILE=/home/oracle/  
pwrsp.properties
```

Related Topics

- [Creating a Password Response File](#)
You can create a password response file and use it with configuration assistants to perform silent installation.

Running Configuration Assistants Using Response Files

You can run configuration assistants in response file or silent mode to configure and start Oracle software after it is installed on the system. To run configuration assistants in response file or silent mode, you must copy and edit a response file template.



Note:

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

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

Related Topics

- [Creating a Password Response File](#)
You can create a password response file and use it with configuration assistants to perform silent installation.

Running Oracle DBCA Using Response Files

You can run Oracle Database Configuration Assistant (Oracle DBCA) in response file mode to configure and start an Oracle database on the system.

To run 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 /directory_path/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 Net Configuration Assistant in silent mode to configure and start an Oracle Net Listener on the system, configure naming methods, and configure Oracle Net service names.

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

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

2. Open the response file in a text editor:

```
$ vi /local_dir/netca.rsp
```

3. Follow the instructions in the file to edit it.

 **Note:**

Net Configuration Assistant fails if you do not correctly configure the response file.

4. Log in as the Oracle software owner user, and set the `ORACLE_HOME` environment variable to specify the correct Oracle home directory.
5. Enter a command similar to the following to run Net Configuration Assistant in silent mode:

```
$ $ORACLE_HOME/bin/netca /silent /responsefile /local_dir/netca.rsp
```

In this command:

- The `/silent` option indicates to run Net Configuration Assistant in silent mode.
- `local_dir` is the full path of the directory where you copied the `netca.rsp` response file template.

B

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 Database 19c with Oracle RAC, all subdirectories except for the Oracle Inventory directory, the Oracle Automatic Storage Management home (if applicable), and the Oracle Clusterware home, are under a top-level Oracle base directory. The Oracle home and `admin` directories are also located under the Oracle base directory.
- [Directory Structures for Oracle RAC](#)
Review this topic to 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 Database 19c with Oracle RAC, all subdirectories except for the Oracle Inventory directory, the Oracle Automatic Storage Management home (if applicable), and the Oracle Clusterware home, are under a top-level Oracle base directory. The Oracle home and `admin` directories are also located under the Oracle base directory.



See Also:

Oracle Database Installation Guide for your platform for more information about the Oracle home and `admin` directories

Directory Structures for Oracle RAC

Review this topic to understand the hierarchical directory tree of a sample Optimal Flexible Architecture (OFA)-compliant database for Oracle RAC.

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

Directory	Description
<code>\$ORACLE_BASE</code>	The default <code>ORACLE_BASE</code> directory, where the software owner is the <code>oracle</code> user. For example: <code>/u01/app/oracle</code>

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

Directory	Description
<i>installation_type</i>	Type of installation under the Oracle base directory. For example, when installing Oracle Database 19c, the value for installation type is <code>product/19.0.0/dbhome_1</code> . For example: <code>/u01/app/oracle/product/19.0.0/dbhome_1</code>
<code>\$ORACLE_HOME</code>	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 19c software, then the path is as follows: <code>/u01/app/oracle/product/19.0.0/dbhome_2</code> Note that the Oracle database home is under the Oracle base directory path. An Oracle Clusterware directory must <i>not</i> be under the Oracle base directory path where the database executable files are located.
<code>db</code>	The directory in which the local initialization parameter file is stored for the database.
<code>admin</code>	The administrative directory. Note that with Oracle Database 11g, <code>bdump</code> , <code>cdump</code> , and <code>udump</code> files are relocated to the directory associated with <code>ADR_BASE</code> . For example: <code>/u01/app/oracle/admin</code>
<i>db_unique_name</i>	The database unique name; this is the same as <code>dbname</code> when the database name is 8 or fewer characters in length. For example, if your database name is <code>sales</code> , the directory path is: <code>/u01/app/oracle/admin/sales</code>
<code>/hdump</code>	The dump destinations for database server.
<code>/pfile</code>	
<code>\$ADR_BASE/bdump</code>	Automatic Diagnostic Repository dump destination trace files.
<code>\$ADR_BASE/cdump</code>	Note that this directory path is set by the initialization parameter <code>DIAGNOSTIC_DEST</code> , and that the path for the Automatic Data Repository must be located on a shared storage location available to all the nodes.
<code>\$ADR_BASE/udump</code>	By default, this path is a subset of the Oracle base directory, in the following path: <code>\$ORACLE_BASE/diag/</code>
Oracle Grid Infrastructure for a cluster home (Grid home)	An OFA-compliant path for the Oracle Clusterware home. The default value is: <code>/u01/app/19.0.0/grid</code> During the Oracle Grid Infrastructure for a cluster installation, Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM) software is installed. The <code>root.sh</code> 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.
<code>bin</code>	The subtree for Oracle Clusterware and Oracle ASM executable files.

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

Directory	Description
<code>network</code>	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

C

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.

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.

- [Using the CVU Database Upgrade Validation Command Options](#)
Use the Cluster Verification Utility (CVU) command `cluvfy stage -pre dbinst -upgrade` to check the readiness of your Oracle RAC installation for upgrades.
- [Example of Verifying System Upgrade Readiness for Oracle RAC](#)
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.
- [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.

Using the CVU Database Upgrade Validation Command Options

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

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.

The command uses the following syntax, where variable content is indicated by italics:

```
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.

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

Option	Description
-src_dbhome <i>src_RAChome</i>	Location of the source Oracle RAC home that you are upgrading, where <i>src_RAChome</i> is the path to the home to upgrade.
-dest_dbhome <i>dest_RAChome</i>	Location of the Oracle RAC home for upgrade, where <i>dest_RAChome</i> is the path to the Oracle RAC home.
-dest_version <i>dest_version</i>	Release number of the upgrade, including any patchset. The release number must include the five digits designating the release to the level of the platform-specific patch. For example: 12.2.0.1.0.
-user <i>user_name</i>	User name to access all the nodes with root privileges.
-location <i>dir_path</i>	Full file system path for the <code>sudo</code> executable.
-dbname <i>db_names_list</i>	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.



See Also:

Oracle Database Administrator's Guide for information about release number format

Example of Verifying System Upgrade Readiness for Oracle RAC

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/18.0.0/dbhome_1  
-dest_dbhome /u01/app/oracle/product/19.0.0/dbhome_1  
-dest_version 19.0.0.0.0 -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

D

Configuring Read-Only 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.
- [Enabling a Read-Only Oracle Home](#)
Configure your Oracle home as a read-only Oracle home after you have performed a software-only Oracle Database installation.
- [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.
- [Determining if an Oracle Home is Read-Only](#)
Run the `orabasehome` command to determine if your Oracle home is a read/write or read-only Oracle home.
- [File Path and Directory Changes in Read-Only Oracle Homes](#)
Examples of hierarchical file mappings in a read-only Oracle home as 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](#)
Starting with Oracle Database 18c, you can configure an Oracle home in read-only mode.
- [About Oracle Base Homes](#)
Both, in a read-only ORACLE_HOME and read/write 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](#)
Both, in a read-only ORACLE_HOME and read/write 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.

About Read-Only Oracle Homes

Starting with Oracle Database 18c, you can configure an Oracle home in read-only mode.

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

In a read-only Oracle home, all the configuration data and log files reside outside of the read-only Oracle home.

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`

**Note:**

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

About Oracle Base Homes

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

In a read/write `ORACLE_HOME`, the `ORACLE_BASE_HOME` path is the same as the `ORACLE_HOME` directory. However, 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/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`.

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

```
$ setenv ORACLE_HOME /u01/app/oracle/product/19.0.0/dbhome_1
$ cd $ORACLE_HOME/bin
$ ./orabasehome
```

For example:

```
$ ./orabasehome
/u01/app/oracle/homes/OraDB19Home1
```

Where, `/u01/app/oracle` is `ORACLE_BASE` and `OraDB19Home1` is `HOME_NAME`

About Oracle Base Config

Both, in a read-only ORACLE_HOME and read/write ORACLE_HOME, the configuration files reside in a location known as ORACLE_BASE_CONFIG.

In a read/write ORACLE_HOME, the ORACLE_BASE_CONFIG path is the same as the ORACLE_HOME path because it is located at \$ORACLE_HOME. However, 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/19.0.0/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.

The `orabasetab` file resides in `ORACLE_HOME/install/orabasetab`. To determine if an ORACLE_HOME is read-only or read/write, you can 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 N in a read/write ORACLE_HOME and 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.

In a read/write ORACLE_HOME, ORACLE_HOME, ORACLE_BASE_HOME and ORACLE_BASE_CONFIG are all the same.

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/19.0.0/dbhome_1/install
```

3. View the contents of the `orabasetab` file.

```
$ cat orabasetab
/u01/app/oracle/product/19.0.0/dbhome_1:/u01/app/
oracle:Oradbl9Home1:Y:
```

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

Enabling a Read-Only Oracle Home

Configure your Oracle home as a read-only Oracle home after you have performed a software-only Oracle Database installation.

To enable a read-only Oracle home:

1. Perform a software-only Oracle Database installation.
2. Run the `roohctl -enable` script.
3. Run Oracle Database Configuration Assistant (Oracle DBCA) to create a database.

Software-Only Database Installation

1. Log in as the Oracle installation owner user account (`oracle`) that you want to own the software binaries.
2. Download the Oracle Database installation image files (`db_home.zip`) to a directory of your choice. For example, you can download the image files to the `/tmp` directory.
3. Create the Oracle home directory and extract the image files that you have downloaded in to this Oracle home directory. For example:

```
$ mkdir -p /u01/app/oracle/product/19.0.0/dbhome_1
$ chown oracle:oinstall /u01/app/oracle/product/19.0.0/dbhome_1
$ cd /u01/app/oracle/product/19.0.0/dbhome_1
$ unzip -q /tmp/db_home.zip
```

 **Note:**

Ensure 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.

4. From the Oracle home directory, run the `runInstaller` command to start the Oracle Database installer.
5. In the Select Configuration Option screen, select **Set Up Software Only**.
6. Select your installation type. Installation screens vary depending on the installation option you select. Respond to the configuration prompts as needed.

 **Note:**

Click **Help** if you have any questions about the information you are asked to submit during installation.

Run the `roohctl` Script

1. Go to the `bin` directory

```
$ cd /u01/app/oracle/product/19.0.0/dbhome_1/bin
```

2. Run the `roohctl` script to enable read-only Oracle home.

```
$ ./roohctl -enable
```

3. On Oracle Real Application Clusters (Oracle RAC) installations, run the preceding `roohctl` script on every node of the cluster. Alternatively, run the `roohctl` script with the `nodelist` option and provide the list of cluster nodes:

```
$ ./roohctl -enable -nodelist comma_separated_list_of_nodes
```

Run Oracle Database Configuration Assistant

1. Ensure that you are still in the `bin` directory and run Oracle DBCA.

```
$ ./dbca
```

2. In the Select Database Operation screen, select **Create a Database**.
3. The configuration screens vary depending on the options you select. Respond to the prompts as needed.

 **Note:**

Click **Help** if you have any questions about the information you are asked to submit during database creation.

Related Topics

- *Oracle Database 2 Day DBA*

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*

Determining if an Oracle Home is Read-Only

Run the `orabasehome` command to determine if your Oracle home is a read/write or read-only Oracle home.

If the output of the `orabasehome` command is the same as `$ORACLE_HOME`, then your Oracle home is in read/write mode. If the output displays the path `ORACLE_BASE/homes/HOME_NAME`, then your Oracle home is in read-only mode.

1. Set the `ORACLE_HOME` environment variable:

Bourne, Bash or Korn shell:

```
$ ORACLE_HOME=/u01/app/oracle/product/19.0.0/dbhome_1  
$ export ORACLE_HOME
```

C shell:

```
% setenv ORACLE_HOME /u01/app/oracle/product/19.0.0/dbhome_1
```

2. Go to the `bin` directory and run the `orabasehome` command:

```
$ cd $ORACLE_HOME/bin
$ ./orabasehome
/u01/app/oracle/homes/OraDB19Home1
```

In this example, the Oracle home is in read-only mode.

File Path and Directory Changes in Read-Only Oracle Homes

Examples of hierarchical file mappings in a read-only Oracle home as compared to a read/write 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 `OraDB19Home1`.

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 D-1 read/write and Read-Only Oracle Home File Path Examples

Directory	Read/Write Oracle Home File Path	Read-Only Oracle Home File Path
ORACLE_HOME	/u01/app/oracle/ product/19.0.0/ dbhome_1	/u01/app/oracle/ product/19.0.0/ dbhome_1
ORACLE_BASE	/u01/app/oracle/	/u01/app/oracle/
ORACLE_BASE_HOME	ORACLE_HOME (or) /u01/app/oracle/ product/19.0.0/ dbhome_1	ORACLE_BASE/homes/ HOME_NAME (or) /u01/app/oracle/homes/ OraDB19Home1
ORACLE_BASE_CONFIG	ORACLE_HOME (or) /u01/app/oracle/ product/19.0.0/ dbhome_1	ORACLE_BASE (or) /u01/app/oracle/
network	ORACLE_HOME/network (or) /u01/app/oracle/ product/19.0.0/ dbhome_1/network	ORACLE_BASE_HOME/network (or) /u01/app/oracle/homes/ OraDB19Home1/network

Table D-1 (Cont.) read/write and Read-Only Oracle Home File Path Examples

Directory	Read/Write Oracle Home File Path	Read-Only Oracle Home File Path
dbs	ORACLE_HOME/dbs (or) /u01/app/oracle/ product/19.0.0/ dbhome_1/dbs	ORACLE_BASE/dbs (or) /u01/app/oracle/dbs

E

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.
- [About Viewing Port Numbers and Access URLs](#)
In most cases, the port number of an Oracle Database component is displayed in the tool used to configure the 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.

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.

About Viewing Port Numbers and Access URLs

In most cases, the port number of an Oracle Database component is displayed in the tool used to configure the port.

In addition, ports for some Oracle Database applications are listed in the `portlist.ini` file. This file is located in the `$ORACLE_HOME/install` directory.

If you change a port number after installation, that port number is not updated in the `portlist.ini` file. For this reason, the `portlist.ini` file is an accurate record of ports configured at the time of installation.

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 E-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

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

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

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

Component	Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
Oracle HA Services daemon (OHASD)	The Oracle High Availability Services (OHAS) daemon starts the Oracle Clusterware stack.	42424	Dynamic	UDP	Yes
Oracle Net Listener	Allows Oracle client connections to the database by using Oracle Net Services. You can configure this port during installation. To reconfigure this port, use NETCA.	1521	Port number changes to the next available port. Modifiable manually to any available port.	TCP	No
Oracle Notification Services (ONS)	Port for ONS, used to publish and subscribe service for communicating information about Fast Application Notification (FAN) events. The FAN notification process uses system events that Oracle Database publishes when cluster servers become unreachable or if network interfaces fail. Use <code>srvctl</code> 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 <i>cannot</i> view it afterward. Refer to <i>Oracle XML DB Developer's Guide</i> for information about changing this port number.	0	Configured manually	FTP	No

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

Component	Description	Default Port Number	Port Range	Protocol	Used Only On Interconnect
Oracle XML DB (HTTP)	<p>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 <i>cannot</i> view it afterward.</p> <p>Refer to <i>Oracle XML DB Developer's Guide</i> for information about changing this port number.</p>	0	Configure manually	HTTP	No
Oracle Trace File Analyzer (TFA)	<p>The Oracle Trace File Analyzer (TFA) daemons in a cluster communicate securely over ports 5000 to 5005.</p> <p>If the port range is not available on your system, then replace it with the ports available on your system using the</p> <pre>tfactl set port</pre> <p>command.</p>	5000	5000 to 5005	SSL/TLS	No

Index

A

- apply patches during install
 - apply patches during upgrade, [2-6](#)
- Automatic Memory Management, [3-4](#)
 - about, [1-3](#)
- automatic undo management, [6-7](#)

C

- candidate disks, [3-10](#)
- CDBs, [2-3](#), [4-4](#)
 - character sets, [3-5](#)
- changing host names, [6-2](#)
- character sets, [3-5](#)
- cluster database
 - installed configuration, [6-5](#)
- Cluster Verification Utility
 - DBCA database creation stage readiness check, [3-7](#)
- components
 - created when using DBCA, [6-5](#)
- connect descriptors, [6-9](#)
- connection load balancing, [6-16](#)
- control files
 - described, [6-6](#)
- converting
 - to Oracle Real Application Clusters from single-instance Oracle databases, [A-1](#)
- copying demo directory, [D-6](#)
- create database
 - using scripts, [A-2](#)
- creating
 - Oracle Real Application Clusters database with Database Configuration Assistant, [3-10](#)
- cron jobs, [1-3](#)

D

- data files
 - and DBCA, [6-5](#)
 - described, [6-5](#)

- database
 - components, created when using DBCA, [6-5](#)
 - services, [6-9](#)
- Database Agent
 - and listeners, [6-22](#)
- Database Agent process, [6-13](#), [6-22](#)
- Database Configuration Assistant
 - components created by, [6-5](#)
 - control files, [6-6](#)
 - creating Oracle Real Application Clusters database
 - after installation, [3-10](#)
 - data files, [6-5](#)
 - deleting Oracle Real Application Clusters databases, [3-14](#)
 - initialization parameter files, [6-7](#)
 - Initialization Parameters page, [3-11](#)
 - List of Cluster Databases page, [3-14](#)
 - Oracle ASM Disk Groups page, [3-10](#)
 - rollback segments, [6-7](#)
 - running in silent mode, [A-16](#)
 - Summary dialog, [3-14](#)
 - tablespaces, [6-5](#)
 - Welcome page, [3-10](#)
- database creation using scripts, [A-2](#)
- DBCA, [6-19](#)
 - configuring Automatic memory Management, [3-4](#)
 - delete database, [3-14](#)
 - loading SSH keys into memory to start, [3-8](#)
 - See also Oracle Database Configuration Assistant (DBCA).
- dbca.rsp file, [A-6](#)
- DBSNMP user
 - password requirements, [3-4](#)
- dedicated servers, [6-17](#)
- deinstall, [7-1](#), [7-3](#)
 - See also removing Oracle software
- deinstall command, [7-1](#)
 - example, [7-6](#)
- deinstallation, [7-1](#)
 - examples, [7-5](#)
- delete database with DBCA, [3-14](#)
- demo directory, [D-6](#)

deprecated features
 service management, [3-6](#)
 directory structure, [B-1](#)

E

enterprise.rsp file, [A-6](#)
 error messages
 ORA-00845, [3-11](#)
 ORA-01078, [3-11](#)
 executeConfigTools, [A-12](#)

F

failover
 and service registration, [6-17](#)
 file paths, [D-9](#)
 files
 dbca.rsp, [A-6](#)
 enterprise.rsp, [A-6](#)
 response files, [A-6](#)
 Free server pool, [5-3](#)
 described, [5-3](#)

G

generic server pool
 described, [5-3](#)
 Generic server pool, [5-3](#)
 global database names
 selecting, [3-3](#)
 globalization, [1-3](#)

H

high availability
 SCAN listeners, [6-11](#)
 host names
 changing, [6-2](#)
 HP-UX user
 setting ownership, [4-3](#)

I

image
 install, [2-1](#)
 initialization parameter files, [6-7](#)
 listeners
 parameters, [6-13](#), [6-22](#)
 initialization parameters
 DISPATCHERS, [6-17](#)
 MEMORY_TARGET, [3-11](#)
 REMOTE_LISTENER, [6-13](#)

installation
 directory structure, [B-1](#)
 listener.ora file, [6-13](#)
 non-interactive, [A-2](#)
 of additional products after installation is
 completed, [3-11](#)
 response files, [A-6](#)
 preparing, [A-6](#), [A-8](#)
 templates, [A-6](#)
 silent mode, [A-9](#)
 tnsnames.ora file, [6-19](#)
 installation option
 Automatic Memory Management, [3-4](#)
 installer screen
 Select Configuration Option, [2-4](#)
 invalid objects
 recompiling, [4-3](#)

L

licensing, [1-3](#)
 List of Cluster Databases page, [3-14](#)
 listener
 automatic migration from 11.2, 12.1, 12.2, or
 18c to 19c, [3-7](#)
 listeners
 default configuration of listener.ora, [6-13](#)
 files
 listener.ora, [6-13](#), [6-22](#)
 parameters
 LOCAL_LISTENER, [6-19](#)
 REMOTE_LISTENER, [6-19](#)
 registration, [6-14](#), [6-18](#)
 service registration, [6-24](#)
 load balancing
 and service registration, [6-17](#)
 local listeners, [6-13](#), [6-22](#)
 log file
 how to see the log file during installation, [2-1](#)
 LREG process
 and listener registration, [6-14](#), [6-18](#)
 discovery routine, [6-14](#), [6-18](#)

M

Memory Size (SGA and PGA), [3-11](#)
 MEMORY_TARGET, [3-11](#)
 multitenant container database
 character sets, [3-5](#)
 multitenant container databases
 See CDBs

N

naming methods, [6-9](#)

Net Configuration Assistant (NetCA)
 response files, [A-18](#)
 running at command prompt, [A-18](#)

net service names, [6-20](#)

netca.rsp file, [A-6](#)

network configuration files
 tnsnames.ora, [6-19](#)

networks
 configuration files, [6-13](#)

noninteractive mode
 See response file mode

O

OINSTALL, [6-2](#)

operating system authentication
 to Oracle ASM, [6-2](#)
 to Oracle Database, [6-2](#)

orabasehome, [D-8](#)

orabasetab, [D-3](#)

Oracle ASM
 and candidate disks, [3-10](#)
 Change disk discovery path, [3-10](#)
 response files, [A-5](#)

Oracle base config, [D-3](#)

Oracle base home, [D-2](#)

Oracle Database Configuration Assistant, [A-17](#)
 response file, [A-6](#)

Oracle Database Configuration Assistant (DBCA)
 no longer sets LOCAL_LISTENER and
 REMOTE_LISTENER, [6-19](#)
 redo log files, [6-6](#)

Oracle Database Upgrade Assistant, [3-7](#)

Oracle DBCA, [A-17](#)

Oracle Fleet Patching and Provisioning
 about, [2-7](#)

Oracle Fleet Patching and Provisioning Client,
[2-7](#)

Oracle Fleet Patching and Provisioning Server,
[2-7](#)

Oracle FPP, [2-7](#)

Oracle Label Security
 configuring after installation, [4-7](#)

Oracle Net Configuration Assistant
 response file, [A-6](#)

Oracle ORAchk
 and Upgrade Readiness Assessment, [1-3](#)

Oracle RAC
 software-only install, [2-4](#)

Oracle RAC deployment, [1-1](#)

Oracle Real Application Clusters
 databases, deleting, [3-14](#)
 overview, [3-1](#), [6-1](#)

Oracle Real Application Clusters One Node
 databases, creating, [3-12](#)

Oracle Real Application Clusters One Node (*continued*)
 databases, deleting, [3-14](#)

Oracle Universal Installer
 response files
 list of, [A-6](#)

ORACLE_BASE environment variable
 setting, [4-3](#)

ORACLE_BASE_CONFIG, [D-3](#), [D-9](#)

ORACLE_BASE_HOME, [D-2](#), [D-9](#)

ORACLE_HOME, [D-9](#)

ORACLE_HOME environment variable
 setting, [4-3](#)

ORACLE_SID environment variable
 setting, [4-3](#)

oratab file, [6-4](#)

OSASM, [6-2](#)

OSBACKUPDBA, [6-2](#)

OSDBA, [6-2](#)

OSDBA for ASM, [6-2](#)

OSDGDBA, [6-2](#)

OSKMDBA, [6-2](#)

OSOPER, [6-2](#)

OSOPER for ASM, [6-2](#)

OSRACDBA, [6-2](#)

other changes, [13](#)

P

parallel query server pool, [3-11](#)

pass phrase, [3-8](#)
 for SSH, [3-8](#)

passwords, [3-4](#)

patch updates, [4-2](#)

PDBs, [2-3](#), [4-4](#)

PGA, [3-4](#)
 and memory management, [1-3](#)

pluggable databases
 See PDBs

policy-managed, [3-11](#)

portlist.ini
 location of, [E-2](#)

ports
 how to find, [E-2](#)

postinstallation
 user accounts setup, [4-5](#)

postinstallation -executeConfigTools option, [A-11](#)

postinstallation configToolAllCommands script,
[A-14](#)

proxy realm, [1-3](#)

R

read only Oracle home, [D-4](#)

read-only oracle home, [D-1](#), [D-2](#), [D-9](#)

read-only Oracle home, [D-1](#), [D-3](#), [D-6](#), [D-8](#)

read/write oracle home, [D-9](#)
 reader nodes, [3-2](#)
 recompiling invalid objects, [4-3](#)
 redo log files
 described, [6-6](#)
 installed configuration, [6-6](#)
 release update revisions, [4-2](#)
 release updates, [4-2](#)
 removing Oracle software, [7-1](#)
 examples, [7-5](#)
 response file, [A-17](#)
 response file installation
 preparing, [A-6](#)
 response files
 templates, [A-6](#)
 silent mode, [A-9](#)
 response file mode, [A-4](#)
 about, [A-4](#)
 reasons for using, [A-4](#)
 See also response files, silent mode
 response files, [A-4](#), [A-17](#)
 about, [A-4](#)
 creating with template, [A-6](#)
 dbca.rsp, [A-6](#)
 enterprise.rsp, [A-6](#)
 for Oracle ASM, [A-5](#)
 general procedure, [A-5](#)
 Net Configuration Assistant, [A-18](#)
 netca.rsp, [A-6](#)
 passing values at command line, [A-4](#)
 specifying with Oracle Universal Installer, [A-9](#)
 See also silent mode.
 rollback segments
 described, [6-7](#)
 roohctl -enable, [D-4](#)
 rootcrs.sh, [7-1](#)
 roothas.sh, [7-1](#)

S

SCAN VIP, [6-11](#)
 scripts to create an Oracle Real Application
 Clusters database, [A-2](#)
 seamless patching, [D-1](#)
 security, [4-7](#)
 selecting passwords, [3-4](#)
 server hardware, [1-2](#)
 server parameter files, [6-7](#)
 server pools
 configuration, [5-1](#)
 described, [5-3](#)
 Free, [5-3](#)
 Generic, [5-3](#)
 service registration, [6-14](#), [6-18](#)
 about, [6-24](#)

service registration (*continued*)
 configuring, [6-16](#)
 SGA, [3-4](#)
 and memory management, [1-3](#)
 shared servers, [6-17](#)
 SID
 See system identifiers (SID)
 silent mode
 about, [A-4](#)
 reasons for using, [A-4](#)
 silent mode installation, [A-9](#)
 SSH, [3-8](#)
 Summary dialog, [3-14](#)
 SYS user
 password requirements, [3-4](#)
 sysasm privilege
 storage tasks requiring, [3-10](#)
 system identifier (SID)
 selecting, [3-3](#)
 system privileges
 understanding, [6-2](#)
 SYSTEM tablespace
 description, [6-5](#)
 SYSTEM user
 password requirements, [3-4](#)

T

tablespaces
 and DBCA, [6-5](#)
 expanding for large sorts, [6-5](#)
 SYSTEM, [6-5](#)
 TEMP, [6-5](#)
 TEMP tablespace
 described, [6-5](#)
 TNS_ADMIN
 and the listener.ora file, [6-13](#), [6-22](#)
 setting a path for, [6-24](#)
 tnsnames.ora file, [6-19](#)
 default configuration, [6-19](#)
 tnsnames.ora files
 and VIP addresses, [6-11](#)
 troubleshooting
 cron jobs and installation, [1-3](#)
 log file, [2-1](#)
 typographic conventions, [x](#)

U

undo management, [6-7](#)
 uninstall
 See removing Oracle software
 upgrade
 of existing Oracle Databases, [3-7](#)
 of listener, [3-7](#)

upgrades
 and SCANs, [6-12](#)
upgrading
 and Oracle ORAchk Upgrade Readiness
 Assessment, [1-3](#)
upgrading Oracle RAC, [1-6](#)
user accounts
 postinstallation setup, [4-5](#)

utlrp.sql, [4-3](#)

V

VIP addresses, [6-11](#)