Oracle® Database

2 Day + Real Application Clusters Guide

12*c* Release 1 (12.1)

E17616-17

March 2015

Oracle Database 2 Day + Real Application Clusters Guide 12*c* Release 1 (12.1)

E17616-17

Primary Author: Janet Stern

Contributing Authors: Mark Bauer, Vivian Schupmann, Douglas Williams, Mike Zampiceni

Contributors: Eric Belden, David Brower, Jonathan Creighton, Sudip Datta, Venkatadri Ganesan, Shamik Ganguly, Prabhaker Gongloor, Mayumi Hayasaka, Masakazu Ito, Aneesh Khandelwal, Sushil Kumar, Barb Lundhild, Venkat Maddali, Gaurav Manglik, Markus Michalewicz, Mughees Minhas, Tim Misner, Joe Paradise, Srinivas Poovala, Hanlin Qian, Mark Scardina, Cathy Shea, Jacqueline Sideri, Vijay Sriram, Vishwanath Subrahmannya Sastry, Ara Vagharshakian

Contents

5 Administering Oracle Clusterware Components

6 Managing Workloads Using Dynamic Database Services

9 Adding and Deleting Nodes and Instances

10 Managing Oracle Software and Applying Patches

Index

List of Figures

Preface

Oracle Database 2 Day + Real Application Clusters Guide describes how to install, configure, and administer Oracle Clusterware, Oracle Automatic Storage Management (Oracle ASM), and Oracle Real Application Clusters (Oracle RAC) on a two-node system using the Oracle Linux operating system.

> **Note:** For Linux operating systems other then Oracle Linux, see *Oracle Real Application Clusters Installation Guide for Linux and UNIX*. For other operating systems, see the platform-specific Oracle RAC installation guide.

This guide covers topics that a reasonably knowledgeable Oracle database administrator (DBA) would need to know when moving from managing a single-instance Oracle Database environment to managing an Oracle RAC environment.

Audience

Oracle Database 2 Day + Real Application Clusters Guide is an Oracle RAC database administration guide for DBAs who want to install and use Oracle RAC. This guide assumes you have already read *Oracle Database 2 Day DBA*. This guide is intended for DBAs who:

- Want basic DBA skills for managing an Oracle RAC environment
- Manage Oracle databases for small- to medium-sized businesses

To use this guide, you should be familiar with the administrative procedures described in *Oracle Database 2 Day DBA*.

> **Note:** Some DBAs may be interested in moving the data from their single-instance Oracle Database to their Oracle RAC database. This guide also explains the procedures for doing this.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following in the Oracle Database documentation set:

- *Oracle Real Application Clusters Installation Guide for Linux and UNIX*
- *Oracle Grid Infrastructure Installation Guide for Linux*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Database 2 Day DBA*
- *Oracle Automatic Storage Management Administrator's Guide*

Conventions

The following text conventions are used in this guide:

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

Changes in This Release for Oracle Database 2 Day + Real Application Clusters Guide

This preface lists changes in *Oracle Database 2 Day + Real Application Clusters Guide*.

Changes in Oracle Database 12*c* Release 1 (12.1)

The following are changes in *Oracle Database 2 Day + Real Application Clusters Guide* for Oracle Database 12*c* Release 1 (12.1).

New Features

The following features are new in this release:

- Oracle Enterprise Manager Database Express replaces Oracle Enterprise Manager Database Control (DB Control)

  In previous Oracle Database releases, DB Control was the primary database management tool described in this manual. DB Control is not included in this release.

  See *Oracle Database 2 Day DBA*.

- Oracle Home User Support for Oracle Database

  Starting with Oracle Database 12*c* Release 1 (12.1), Oracle Database supports the use of an Oracle Home User, which can be specified at installation time. The Oracle Home User can be a Built-in Account or a Windows User Account. If you specify a Windows User Account, then the user should be a low privileged (non-Administrator) account to ensure that the Oracle Home User has a limited set of privileges. Using an Oracle Home User ensures that Oracle Database services have only those privileges required to run Oracle products.

  See *Oracle Database Platform Guide for Microsoft Windows*.

- Application Continuity

  Application Continuity attempts to mask outages from end users and applications by recovering requests following recoverable outages, unplanned and planned. Application Continuity performs this recovery beneath the application so that the outage appears to the application as a delayed execution. Application Continuity masks recoverable outages - those in which requests would succeed if they were re-issued. Examples include system failures, network disconnects, foreground failures, and storage failures.

  See *Oracle Database Development Guide*.

- Transaction Guard

Before Oracle Database 12*c* Release 1 (12.1), it was difficult for a database application to recover after an outage, because the commit message that Oracle Database (the server) returned to the application (the client) was not persistent. If the connection between Oracle Database and the application broke or if the database session became unavailable, then the application received a disconnection error message. After an outage, if an application user resubmitted an in-flight transaction that had been committed, then duplicate transactions resulted.

The Oracle Database feature Transaction Guard ensures that each transaction executes at most once. Its PL/SQL interface, the `DBMS_APP_CONT.GET_LTXID_OUTCOME` procedure, enables an application to determine the outcome of the in-flight transaction after an outage and then recover any work that was lost due to the outage.

See *Oracle Database Development Guide*.

- Java and JDBC Support for Application Continuity

  Application Continuity for Java is available with Oracle Database, JDBC Thin driver, and the Oracle connection pools: UCP (Universal Connection Pool) and WebLogic Server Active GridLink.

  Application Continuity is transparent for Java EE and Java SE applications that use Oracle JDBC, use Oracle connection pools (UCP or WLS Active GridLink), and do not have external actions. For applications with external actions (for example, using autonomous transactions or using UTL_HTTP to issue an SOA call), Application Continuity is transparent only if the application's correctness is preserved when these external actions are replayed after a failure.

  See *Oracle Database JDBC Developer's Guide*.

- Oracle JDBC Support for Transaction Guard

  See *Oracle Database JDBC Developer's Guide*.

Desupported Features

Some features previously described in this document are desupported in Oracle Database 12*c* Release 1 (12.1). See *Oracle Database Upgrade Guide* for a list of desupported features.

1

Introduction to Deploying a Real Application Clusters Environment

Oracle Real Application Clusters (Oracle RAC) enables an Oracle database to run across a cluster of servers, providing fault tolerance, performance, and scalability with no application changes necessary. Oracle RAC provides high availability for applications by removing the single point of failure with a single server.

- About the 2 Day + Real Application Clusters Guide
- About Oracle Grid Infrastructure for a Cluster and Oracle RAC
- About Oracle Automatic Storage Management
- About Oracle Real Application Clusters
- Tools for Installing, Configuring, and Managing Oracle RAC

About the 2 Day + Real Application Clusters Guide

This guide is a task-oriented guide for Oracle Real Application Clusters (Oracle RAC) database administration. This guide shows you how to configure and manage the environment for Oracle Clusterware and Oracle RAC. The administration procedures and examples in this guide assume Oracle Linux is the operating system. For the purposes of this guide, a two-node cluster is used in the procedures and examples.

The goal of this guide is to help you understand the basic steps required to install and maintain an Oracle RAC environment, including how to perform basic troubleshooting, performance monitoring, and backup and recovery activities. This guide is based on Oracle Linux, but you do not have to be a Linux expert to use this guide.

What This Guide Is Not

The 2 Day + Real Application Clusters guide is not a comprehensive description of Oracle RAC. It describes concepts only when necessary for completing a particular task.

See Also:

- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*

Related Materials

The 2 Day + Real Application Clusters guide is part of a comprehensive set of learning materials for administering Oracle Databases, which includes a 2 Day DBA Oracle By Example (OBE) series (available on the web) and Oracle University instructor-led classes. The OBE series also has *viewlets*, or animated demos you view using a web browser.

You can view the OBE content for Oracle RAC at the following website:

`http://www.oracle.com/technetwork/tutorials/`

Overview of Oracle Real Application Clusters Documentation

The 2 Day + Real Application Clusters guide describes how to install, configure, and manage Oracle RAC, Oracle Clusterware, and Oracle Automatic Storage Management (Oracle ASM). It provides examples for how you could deploy Oracle RAC on a two-node cluster, using the Oracle Linux operating system. Oracle Linux is fully compatible — both source and binary — with Red Hat Enterprise Linux. It includes the identical set of packages at the same version levels with the same source code as the Red Hat distribution.

This guide is for DBAs who have experience with single-instance Oracle environments and have read *Oracle Database 2 Day DBA*.

Useful Oracle RAC Guides

Oracle provides a comprehensive set of guides for Oracle RAC. The following list contains useful related documentation:

- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX* (or other operating system)
- *Oracle Grid Infrastructure Installation Guide for Linux* (or other operating system)

> **Note:** Additional information for this release may be available in the *Oracle Database Readme* or *Oracle Database Release Notes for Linux* for Oracle Database 12*c* Release 1 (12.1). You can locate these documents on your Oracle product installation media or on the Oracle Technology Network website:
>
> `http://www.oracle.com/technetwork`

Other Useful Oracle Documentation

In addition to the Oracle Database 2 Day DBA guide, there are many other books that are referenced when performing DBA tasks. The following list displays other books in the Oracle documentation library that were referenced within this guide:

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*
- *Oracle Database Net Services Administrator's Guide*

- *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems*
- *Oracle Database Licensing Information*
- *Oracle Database Reference*
- *Oracle Database Backup and Recovery User's Guide*
- *Oracle Database 2 Day + Java Developer's Guide*
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database Performance Tuning Guide*
- *Oracle Universal Installer NextGen Installation Guide*
- *Oracle Database Upgrade Guide*

About Oracle Grid Infrastructure for a Cluster and Oracle RAC

Oracle Clusterware and Oracle ASM are installed into a single home directory, which is called the **Grid home**. Oracle Grid Infrastructure for a cluster refers to the installation of the combined products. Oracle Clusterware and Oracle ASM are still individual products, and are referred to by those names.

Oracle Clusterware enables servers, referred to as **hosts** or **node**s, to operate as if they are one server, commonly referred to as a **cluster**. Although the servers are standalone servers, each server has additional processes that communicate with other servers. In this way the separate servers appear as if they are one server to applications and end users. Oracle Clusterware provides the infrastructure necessary to run Oracle RAC. The combined processing power of the multiple servers provides greater availability, throughput, and scalability than is available from a single server.

The nodes in a cluster can be organized into a **server pool** for better resource management. Each server pool has the following properties:

- The minimum number of nodes that should be in the server pool
- The maximum number of nodes that can be in the server pool
- The relative importance of this server pool to other server pools

Upon installation of Oracle Grid Infrastructure for a cluster, a default server pool, called the **Free pool**, is created automatically. All servers in a new installation are assigned to the Free server pool, initially. If you create a new server pool, then the servers move from the Free pool to the new server pool automatically.

When you create an Oracle RAC database that is a **policy-managed database**, you specify the number of servers that are needed for the database, and a server pool is automatically created for the database. Oracle Clusterware populates the server pool with the servers it has available. If you do not use server pools, then you create an **administrator-managed database**.

> **Note:** This guide discusses the creation and management of only an administrator-managed Oracle RAC database.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Clusterware Administration and Deployment Guide*

About Oracle Automatic Storage Management

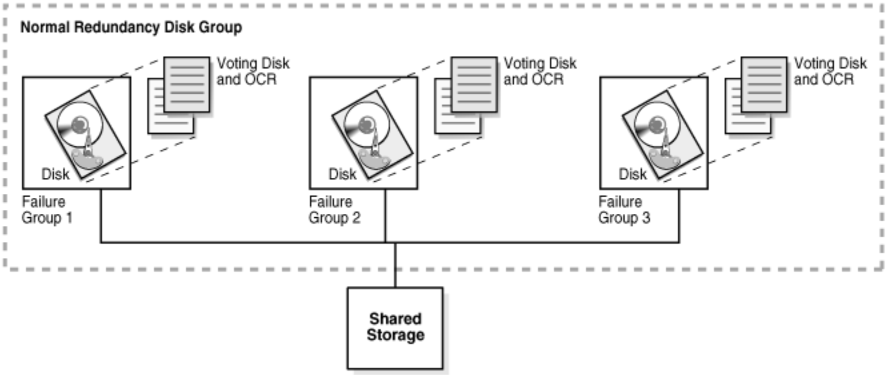
Oracle Automatic Storage Management (Oracle ASM) is an integrated, high-performance volume manager and file system. Oracle ASM supports storing the Oracle Clusterware OCR and voting disk files, and also provides a general purpose cluster file system called Oracle Automatic Storage Management Cluster File System (Oracle ACFS). Oracle ASM is based on the principle that the database should manage storage instead of requiring an administrator to do it. Oracle ASM eliminates the need for you to directly manage potentially thousands of Oracle database files.

About Using Oracle ASM Disk Groups with Oracle RAC

Oracle ASM groups the disks in your storage system into one or more disk groups. You manage a small set of disk groups and Oracle ASM automates the placement of the database files within those disk groups. Oracle Cluster Registry (OCR) and voting disks can also be placed on Oracle ASM disk groups. When using Oracle Real Application Clusters (Oracle RAC), each instance must have access to the data files and recovery files for the Oracle RAC database. Using Oracle ASM is an easy way to satisfy this requirement.

In the following figure, the normal redundancy disk group stored in Oracle ASM contains three failure groups (shown as three disks) to store the voting disk and OCR files for Oracle Clusterware. Because the voting disk records which nodes are currently members of the cluster, there must be at least three voting disks for a two-node cluster. If the Oracle ASM disk group did not contain the voting disk and OCR files, then you could have a normal redundancy disk group with only two failure groups.

Figure 1–1 Oracle Clusterware Files Stored in an Oracle ASM Disk Group



See Also:

- "About Oracle Clusterware" on page 5-1 for information about the Oracle Clusterware files
- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM

Benefits of Using Oracle ASM

Oracle ASM provides the following benefits:

- **Striping**—Oracle ASM spreads data evenly across all disks in a disk group to optimize performance and utilization. This even distribution of database files eliminates the need for regular monitoring and I/O performance tuning.
- **Mirroring**—Oracle ASM increases data availability by optionally mirroring any file. Oracle ASM mirrors at the file level, unlike operating system mirroring, which mirrors at the disk level. Mirroring means keeping redundant copies, or mirrored copies, of each extent of the file, to help avoid data loss caused by disk failures. The mirrored copy of each file extent is always kept on a different disk from the original copy. If a disk fails, then Oracle ASM continues to access affected files by accessing mirrored copies on the surviving disks in the disk group.
- **Online storage reconfiguration and dynamic rebalancing**—When using Oracle ASM, you can add or remove disks from your disk storage system while the database is operating. When you add a disk to a disk group, Oracle ASM automatically redistributes the data so that it is evenly spread across all disks in the disk group, including the new disk. The process of redistributing data so that it is also spread across the newly added disks is known as **rebalancing**. It is done in the background and with minimal impact to database performance.
- **Managed file creation and deletion**—Oracle ASM further reduces administration tasks by enabling files stored in Oracle ASM disk groups to be managed by Oracle Database. Oracle ASM automatically assigns file names when files are created, and automatically deletes files when they are no longer needed by the database.

About Oracle ASM Cluster File System

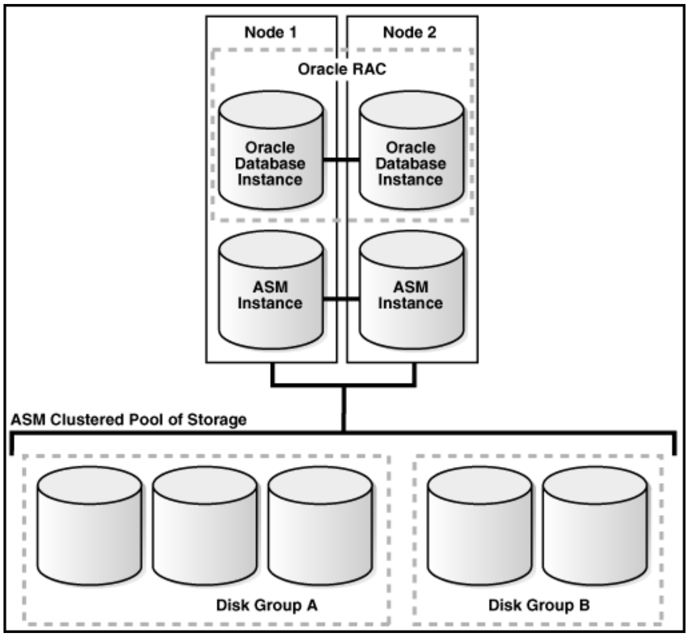
Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a new multi-platform, scalable file system, and storage management technology that extends Oracle ASM functionality to support customer files maintained outside of the Oracle database. Files supported by Oracle ACFS include database and application executables, trace files, alert logs, application reports, BFILEs, and configuration files. Oracle ACFS leverages Oracle ASM functionality to provide dynamic file system resizing, striping, and mirroring.

About Oracle ASM Running in an Oracle RAC Environment

Oracle ASM is implemented as a special kind of Oracle instance, with its own System Global Area and background processes. The Oracle ASM instance is tightly integrated with Oracle Clusterware and Oracle Database. Every server running one or more database instances that use Oracle ASM for storage has an Oracle ASM instance. In an Oracle RAC environment, there is one Oracle ASM instance for each node, and the Oracle ASM instances communicate with each other on a peer-to-peer basis. Only one Oracle ASM instance is supported on a node, but you can have multiple database instances that use Oracle ASM residing on the same node.

The following figure shows the storage configuration commonly used for a 2-node Oracle RAC database.

Figure 1–2 Oracle Instances and Oracle ASM Storage



Oracle recommends that you use Oracle ASM for your Oracle Clusterware files and Oracle RAC datafiles. Oracle databases can use both Oracle ASM files and non-Oracle ASM files. You can also create a file system using Oracle ACFS to store your database Oracle Home and any other external (non-database) files in the cluster.

See Also:

- "About Oracle Clusterware" on page 5-1 for information about the Oracle Clusterware files
- *Oracle Automatic Storage Management Administrator's Guide* for information about Oracle ASM concepts and features

About Oracle Real Application Clusters

Oracle RAC extends Oracle Database so that you can store, update, and efficiently retrieve data using multiple database instances on different servers at the same time. Oracle RAC provides the software that manages multiple servers and instances as a single group. The data files that comprise the database must reside on shared storage that is accessible from all servers that are part of the cluster. Each server in the cluster runs the Oracle RAC software.

An Oracle database has a one-to-one relationship between data files and the database instance. An Oracle RAC database, however, has a one-to-many relationship between data files and database instances. In an Oracle RAC database, multiple instances access a single set of database files.

Each database instance in an Oracle RAC database uses its own memory structures and background processes. Oracle RAC uses Cache Fusion to synchronize the data stored in the buffer cache of each database instance. Cache Fusion moves current data blocks (which reside in memory) between database instances, rather than having one database instance write the data blocks to disk and requiring another database instance to reread the data blocks from disk. When one instance requires a data block located in the buffer cache of another instance, Cache Fusion transfers the data block directly between the instances using the **interconnect**, enabling the Oracle RAC database to access and modify data as if the data resided in a single buffer cache.

Oracle RAC is also a key component for implementing the enterprise grid computing architecture using Oracle software. Having multiple database instances accessing a single set of data files prevents the server from being a single point of failure. If a node in the cluster fails, then the Oracle database continues running on the remaining nodes. Individual nodes can be shutdown for maintenance while application users continue to work.

Oracle RAC supports mainstream business applications, such as OLTP, DSS, and also popular packaged products such as SAP, PeopleSoft, Siebel, and Oracle E*Business Suite, and custom applications. Any packaged or custom application that scales on an Oracle database scales well on Oracle RAC without requiring changes to the application code.

You will learn more about the operation of the Oracle RAC database in a cluster, how to build the cluster, and the structure of an Oracle RAC database in other sections of this guide.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about Oracle RAC concepts and architecture
- *Oracle Clusterware Administration and Deployment Guide* for more information about Oracle Clusterware concepts and architecture

About Oracle RAC One Node

Oracle Real Application Clusters One Node (Oracle RAC One Node) is a single instance of an Oracle RAC database that runs on one node in a cluster. This feature enables you to consolidate many databases into one cluster with minimal overhead, protecting them from both planned and unplanned downtime. The consolidated databases reap the high availability benefits of failover protection, online rolling patch application, and rolling upgrades for the operating system and Oracle Clusterware.

Oracle RAC One Node enables better availability than cold failover for single-instance databases because of the Oracle technology called online database relocation, which intelligently migrates database instances and connections to other cluster nodes for high availability and load balancing. Online database relocation is performed using the Server Control Utility (SRVCTL).

If you run your applications on Oracle RAC One Node, and your applications grow to the point that a single node cannot supply the resources they need, then Oracle RAC One Node can be upgraded online to Oracle Real Application Clusters. If the node running your Oracle RAC One Node database becomes overloaded, then you can migrate the database instance to another node in the cluster using online database relocation with no downtime for application users

See Also:

- "About Oracle Real Application Clusters"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about Oracle RAC One Node

About Installing Oracle RAC on Different Operating Systems

If you plan to install and configure Oracle RAC on an operating system other than Oracle Linux, then you can still use this guide to obtain a general understanding about how to deploy Oracle RAC. You can also use this guide for deploying Oracle RAC on clusters with more than two nodes. For all environments that do not match the environment that this guide describes, modify the examples accordingly.

When installing Oracle RAC on a different platform or different operating system version than Oracle Linux or Red Hat Enterprise Linux, refer to the installation and configuration guides for that platform. For example, if you are installing Oracle RAC on the Oracle Solaris operating system, then you would use the following guides:

- *Oracle Grid Infrastructure Installation Guide for Oracle Solaris*
- *Oracle Real Application Clusters Installation Guide for Solaris Operating System*

Oracle Clusterware and Oracle RAC do not support heterogeneous platforms in the same cluster. For example, you cannot have one node in the cluster running Oracle Linux and another node in the same cluster running Oracle Solaris. All nodes must run the same operating system; that is, they must be binary compatible. Oracle RAC does not support machines having different chip architectures in the same cluster. However, you can have machines of different speeds and sizes in the same cluster.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* for an overview of installing Oracle RAC

Tools for Installing, Configuring, and Managing Oracle RAC

This section describes the tools used for installing, configuring, and managing Oracle Clusterware, Oracle Automatic Storage Management (Oracle ASM), and Oracle Real Application Clusters (Oracle RAC).

Table 1–1 Tools for Installing, Configuring, and Managing Oracle RAC

Tool or Utility	Purpose	Description of Its Use
Oracle Universal Installer (OUI)	Installs Oracle Grid Infrastructure for a cluster (which consists of Oracle Clusterware and Oracle ASM) and Oracle Database with Oracle RAC.	After you configure the nodes to use in your cluster, OUI installs the Oracle software on the specified nodes using a network connection.
Cluster Verification Utility (CVU)	Validates a range of cluster and Oracle RAC components such as shared storage devices, networking configurations, system requirements, and Oracle Clusterware, also operating system groups and users.	CVU is a command-line tool that you can use for preinstallation and postinstallation checks of your cluster environment. CVU is especially useful during preinstallation and during installation of Oracle Clusterware and Oracle RAC components. OUI runs CVU after the Oracle Clusterware installation to verify your environment.

Table 1–1 (Cont.) Tools for Installing, Configuring, and Managing Oracle RAC

Tool or Utility	Purpose	Description of Its Use
Oracle Enterprise Manager	The primary product for managing your database	Oracle Enterprise Manager Cloud Control is a graphical user interface (GUI) that you use to manage single-instance and Oracle RAC databases, Oracle Clusterware, and Oracle ASM. After you have installed the Oracle Database software, created or upgraded a database, and configured the network, you can use Oracle Enterprise Manager to manage your database. Database Control also provides an interface for performance advisors and for Oracle utilities such as SQL*Loader and Recovery Manager (RMAN).
SQL*Plus	Issues SQL and PL/SQL statements to Oracle Database	SQL*Plus provides a command-line interface that enables you to perform database management operations for a database, and also to query, insert, update, or delete data directly in the database.
Server Control (SRVCTL)	Manages the resources defined in the Oracle Cluster Registry (OCR). These resources include the node applications, called **nodeapps**, that comprise Oracle Clusterware, which includes the Oracle Notification Service (ONS), and the Virtual IP (VIP).	You use SRVCTL, a command-line interface, to manage databases, instances, listeners, services, and applications that run in an Oracle Clusterware environment. Using SRVCTL you can start and stop nodeapps, databases, instances, listeners, and services, delete or move instances and services, add services, and manage configuration information.
Cluster Ready Services Control (CRSCTL)	Manages Oracle Clusterware daemons. These daemons include Cluster Synchronization Services (CSS), Cluster-Ready Services (CRS), and Event Manager (EVM).	CRSCTL is a command-line tool that you can use to start and stop Oracle Clusterware and to determine the current status of your Oracle Clusterware installation.
Database Configuration Assistant (DBCA)	Creates and configures Oracle Databases	DBCA can be launched by OUI, depending upon the type of install that you select. You can also launch DBCA as a standalone tool at any time after Oracle Database installation. You can run DBCA in interactive mode or noninteractive/silent mode. Interactive mode provides a graphical interface and guided workflow for creating and configuring a database. DBCA is the preferred way to create a database, because it is a more automated approach, and your database is ready to use when DBCA completes.
Oracle Automatic Storage Management Configuration Assistant (ASMCA)	Installs and configures Oracle ASM	ASMCA is a utility that supports installing and configuring Oracle ASM instances, disk groups, volumes, and Oracle Automatic Storage Management Cluster File System (Oracle ACFS). ASMCA provides both a GUI and a non-GUI interface.

Table 1–1 (Cont.) Tools for Installing, Configuring, and Managing Oracle RAC

Tool or Utility	Purpose	Description of Its Use
Oracle Automatic Storage Management Command Line utility (ASMCMD)	Administers Oracle ASM	ASMCMD is a command-line utility that you can use to manage Oracle ASM instances, Oracle ASM disk groups, file access control for disk groups, files and directories within Oracle ASM disk groups, templates for disk groups, and Oracle ASM volumes.
Listener Control (LSNRCTL)	Administers listeners	The Listener Control utility is a command-line interface that you use to administer listeners.You can use its commands to perform basic management functions on one or more listeners. Additionally, you can view and change parameter settings for the listener.

If you have installed Oracle Grid Infrastructure for a cluster for Oracle Database 12*c* Release 1 (12.1), then, when using utilities to manage your cluster, databases, database instances, Oracle ASM, and listeners, use the appropriate binary that is in the home directory of the object or component you are managing. Set the `ORACLE_HOME` environment variable to point to this directory, for example:

- If you use ASMCMD, SRVCTL, SQL*Plus, or LSNRCTL to manage Oracle ASM or its listener, then use the executable files located in the Grid home, not the executable files located in the Oracle Database home, and set the `ORACLE_HOME` environment variable to the location of the Grid home.
- If you use SRVCTL, SQL*Plus, or LSNRCTL to manage a database instance or its listener, then use the binaries located in the Oracle home where the database instance or listener is running, and set the `ORACLE_HOME` environment variable to the location of that Oracle home

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about the tools and utilites used to manage Oracle RAC

2

Preparing Your Cluster

You (the DBA) must work with your system administrator and network administrator to configure the two nodes in your cluster before you install the Oracle Real Application Clusters software.

> **Note:** These topics assume a basic understanding of the Linux operating system. In some cases, you may need to refer to details in *Oracle Grid Infrastructure Installation Guide* for your operating system.
>
> In addition, you must have `root` or `sudo` privileges to perform certain tasks (or Administrator privileges on Windows systems).

- Verifying System Requirements
- Preparing the Server
- Configuring the Network
- Preparing the Operating System and Software
- Configuring Installation Directories and Shared Storage

See Also:

- "About Installing Oracle RAC on Different Operating Systems"
- "Tools for Installing, Configuring, and Managing Oracle RAC"

Verifying System Requirements

Before you begin your installation, check to ensure that your system meets the requirements for Oracle Real Application Clusters (Oracle RAC). The requirements can be grouped into the following categories:

- Checking Operating System Certifications
- About Hardware Requirements
- About Network Hardware Requirements
- About IP Address Requirements
- Verifying Operating System and Software Requirements

See Also:

- "About Installing Oracle RAC on Different Operating Systems"
- "Preparing the Server"
- "Configuring Installation Directories and Shared Storage"
- Oracle Support Note [ID 964664.1]: "Oracle Product Lines - Tips for finding Certification Information"

Checking Operating System Certifications

You must ensure that you have a certified combination of the operating system and the Oracle Database software. Refer to My Oracle Support certification, which is located at the following website

`https://support.oracle.com`

You can find certification information by selecting the **Certifications** tab. You can also search for support notes that contain instructions on how to locate the Certification information for your platform.

Note: Oracle Universal Installer verifies that your server and operating system meet the listed requirements. However, you should check the requirements before you start Oracle Universal Installer to ensure your server and operating system meet the requirement. This helps to avoid delays in the software installation process that you might incur if your hardware or software is not certified.

See Also:

- "Preparing the Server"
- "Verifying Operating System and Software Requirements"

About Hardware Requirements

Each node you make part of your cluster, or Oracle Clusterware and Oracle RAC installation, must satisfy the minimum hardware requirements of the software.

Note: When you install Oracle software, Oracle Universal Installer (OUI) automatically performs hardware prerequisite checks and notifies you if they are not met.

The minimum hardware requirements are:

- At least 4 GB of RAM for Oracle Grid Infrastructure for a Cluster installations, including installations where you plan to install Oracle RAC
- An amount of swap space that is at least equal to the amount of RAM
- Temporary space (at least 1 GB) available in `/tmp`
- A processor type (CPU) that is certified with the release of the Oracle software being installed (64-bit)

- A minimum of 1024 x 786 display resolution, so that Oracle Universal Installer (OUI) displays correctly
- All servers in the cluster have the same chip architecture, for example, all 64-bit processors
- Adequate disk space in the software installation locations to store the Oracle software:

  Note: Refer to the *Oracle Grid Infrastructure Installation Guide* and the *Oracle Real Application Clusters Installation Guide* for your operating system for the actual disk space requirements. The amount of disk space used by the Oracle software can vary, and might be higher than what is listed in this guide.

  – At least 8 GB of available disk space for the *Grid home* directory, which includes both the binary files for Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM)

  – Additional disk space (100 GB is recommended) for the *Grid home* directory for associated log files and patches.

  – At least 3.5 GB of space for the Oracle base of the Oracle Grid Infrastructure installation owner (Grid user). The Oracle base includes Oracle Clusterware and Oracle ASM log files.

  – At least 5.8 GB of disk space for the Oracle home (the location for the Oracle Database software binaries).

  See Also:

  - *Oracle Grid Infrastructure Installation Guide* for your platform
  - "About Performing Platform-Specific Configuration Tasks"
  - "Preparing the Server"
  - "Configuring Installation Directories and Shared Storage"

About Shared Storage

An Oracle RAC database is a *shared everything* database. All data files, control files, redo log files, and the server parameter file (SPFILE) used by the Oracle RAC database must reside on shared storage that is accessible by all the Oracle RAC database instances. The Oracle RAC installation demonstrated in this guide uses Oracle ASM for the shared storage for Oracle Clusterware and Oracle Database files.

Oracle Clusterware achieves superior scalability and high availability by using the following components:

- **Voting disk**–Manages cluster membership and arbitrates cluster ownership between the nodes in case of network failures. The voting disk is a file that resides on shared storage. For high availability, Oracle recommends that you have multiple voting disks, and that you have an odd number of voting disks. If you define a single voting disk, then use mirroring at the file system level for redundancy.
- **Oracle Cluster Registry (OCR)**–Maintains cluster configuration information and configuration information about any cluster database within the cluster. The OCR contains information such as which database instances run on which nodes and

which services run on which databases. The OCR also stores information about processes that Oracle Clusterware controls. The OCR resides on shared storage that is accessible by all the nodes in your cluster. Oracle Clusterware can multiplex, or maintain multiple copies of, the OCR and Oracle recommends that you use this feature to ensure high availability.

> **Note:** If you choose not to use Oracle ASM for storing your Oracle Clusterware files, then both the voting disks and the OCR must reside on a cluster file system that you configure before you install Oracle Clusterware in the Grid home.

These Oracle Clusterware components require the following disk space on a shared file system:

- Three Oracle Clusterware Registry (OCR) files, a least 400 MB for each, or 1.2 GB total disk space
- Three voting disk files, 300 MB each, or 900 MB total disk space

If you are not using Oracle ASM for storing Oracle Clusterware files, then for best performance and protection, you should use multiple disks, each using a different disk controller for voting disk file placement. Ensure that each voting disk is configured so that it does not share any hardware device or have a single point of failure.

> **Note:** Oracle recommends that you create a Grid Infrastructure Management Repository. This repository is an optional component, but if you do not select this feature during installation, then you lose access to Oracle Database Quality of Service management, Memory Guard, and Cluster Health Monitor. You cannot enable these features after installation except by reinstalling Oracle Grid Infrastructure.
>
> If you choose to create a Grid Infrastructure Management Repository, then additional space may be needed for shared storage. See the chapter on configuring storage in *Oracle Grid Infrastructure Installation Guide*.

See Also:

- *Oracle Grid Infrastructure Installation Guide* for your platform
- "Configuring Installation Directories and Shared Storage"
- "Configuring Shared Storage"

About Network Hardware Requirements

Oracle Clusterware requires that you connect the nodes in the cluster to a private network by way of a private **interconnect**. The private interconnect is a separate network that you configure between cluster nodes. The interconnect serves as the communication path between nodes in the cluster. This interconnect should be a *private* interconnect, meaning it is *not* accessible to nodes that are not members of the cluster.

The interconnect used by Oracle RAC is the same interconnect that Oracle Clusterware uses. Each cluster database instance uses the interconnect for messaging to synchronize the use of shared resources by each instance. Oracle RAC also uses the interconnect to transmit data blocks that are shared between the instances.

When you configure the network for Oracle RAC and Oracle Clusterware, each node in the cluster must meet the following requirements:

- Each node must have at least two network interface cards (NICs), or **network adapters**. One adapter is for the public network interface and the other adapter is for the private network interface (the interconnect). Install additional network adapters on a node if that node meets either of the following conditions:
  - Does not have at least two network adapters
  - Has two network interface cards but is using network attached storage (NAS). You should have a separate network adapter for NAS.
  - Has two network cards, but you want to use the **Redundant Interconnect Usage** feature, which allows multiple network adapters to be accessed as a single adapter. Redundant Interconnect Usage enables load-balancing and high availability across multiple (up to 4) private networks (also known as interconnects).

  See Also: *Oracle Grid Infrastructure Installation Guide for Linux* for more information about the configuration requirements for Redundant Interconnect Usage

- For easier management, public interface names should be the same for all nodes. If the public interface on one node uses the network adapter `eth0`, then you should configure `eth0` as the public interface on all nodes. Network interface names are case-sensitive.
- The network adapter for the public interface must support TCP/IP.
- The network adapter for the private interface must support the user datagram protocol (UDP) using high-speed network adapters and a **network switch** that supports TCP/IP (Gigabit Ethernet or better).

  Note:

  - UDP is the default interface protocol for Oracle RAC and Oracle Clusterware.
  - You must use a switch for the interconnect. Oracle recommends that you use a dedicated network switch. Token-rings or crossover cables are not supported for the interconnect.
  - Loopback devices are not supported.

- For the private network, the endpoints of all designated interconnect interfaces must be completely reachable on the network. Every node in the cluster must be able to connect to every private network interface in the cluster.
- The host name of each node must conform to the RFC 952 standard, which permits alphanumeric characters. Host names using underscores ("_") are not allowed.

  See Also:

  - "Configuring the Network"
  - "Verifying System Requirements"

About IP Address Requirements

When performing an advanced installation of Oracle Grid Infrastructure for a cluster, you can chose to use **Grid Naming Service (GNS)** and **Dynamic Host Configuration Protocol (DHCP)** for virtual IPs (VIPs). GNS uses **multicast Domain Name Server (mDNS)** to enable the cluster to assign host names and IP addresses dynamically as nodes are added and removed from the cluster, without requiring additional network address configuration in the domain name server (DNS). For more information about GNS, refer to *Oracle Grid Infrastructure Installation Guide* for your platform.

You can configure the public network interfaces for cluster nodes to use IPv4, IPv6, or both types of IP addresses. During installation, the nodes can be connected to networks that have both address types, but you cannot configure the VIPs to use both address types during installation. After installation, you can configure cluster member nodes with a mixture of IPv4 and IPv6 addresses. Oracle Grid Infrastructure and Oracle RAC support the standard IPv6 address notations specified by RFC 2732. Refer to *Oracle Grid Infrastructure Installation Guide* for your platform for addition information and restrictions.

This guide documents how to perform a typical installation, which does not use GNS and uses only IPv4 addresses. You must configure the following addresses manually in your corporate DNS:

- A fixed public IP address for each node
- A virtual IP address for each node
- Three single client access name (SCAN) addresses for the cluster

Note: Oracle Clusterware uses interfaces marked as private as the cluster interconnects. You do not need to specify IP addresses for the interconnect.

During installation of the Oracle Grid Infrastructure for a cluster, a **SCAN** for the cluster is configured, which is a domain name that resolves to all the SCAN addresses allocated for the cluster. The public IP addresses, SCAN addresses, and VIP addresses must all be on the same subnet. The SCAN must be unique within your network. The SCAN addresses should not respond to `ping` commands before installation. In a Typical installation, the SCAN is also the name of the cluster, so the SCAN must meet the same requirements as the cluster name and can be no longer than 15 characters.

During installation of the Oracle Grid Infrastructure for a cluster, a listener is created for each of the SCAN addresses. Clients that access the Oracle RAC database should use the SCAN or SCAN address, not the VIP name or address. If an application uses a SCAN to connect to the cluster database, then the network configuration files on the client computer do not have to be modified when nodes are added to or removed from the cluster. The SCAN and its associated IP addresses provide a stable name for clients to use for connections, independent of the nodes that form the cluster. Clients can connect to the cluster database using the easy connect naming method and the SCAN.

See Also:

- "Configuring the Network"
- "Verifying System Requirements"
- *Oracle Database Net Services Administrator's Guide* for information about the easy connect naming method
- *Oracle Grid Infrastructure Installation Guide* for your platform for more information about configuring GNS and IPv6 network addresses for your cluster

Verifying Operating System and Software Requirements

Refer to the *Oracle Grid Infrastructure Installation Guide* and the *Oracle Real Application Clusters Installation Guide* for your platform for information about specific requirements for your environment.

About Operating System and Software Requirements

The operating system and software requirements might include:

- The operating system release
- The kernel version of the operating system
- Modifying the values for kernel parameters
- Installed packages, patches, or patch sets
- Installed compilers and drivers
- Web browser type and version
- Additional application software requirements

If you are currently running an operating system release that is not supported by Oracle Database 12*c* Release 1 (12.1), then you must first upgrade your operating system before installing Oracle Real Application Clusters.

If you are using Oracle Linux as your operating system, then you can use the Oracle Validated RPM system configuration script to configure your system.

About Installation Fixup Scripts

Oracle Universal Installer (OUI) detects when the minimum requirements for an installation are not met, and creates shell scripts, called Fixup scripts, to finish incomplete system configuration steps. If OUI detects an incomplete task, then it generates a Fixup script (`runfixup.sh`). You can run the script after you click **Fix and Check Again**.

Fixup scripts do the following:

- If necessary set, kernel parameters to values required for successful installation, including:
  - Shared memory parameters
  - Open file descriptor and UDP send/receive parameters
- Create and set permissions on the Oracle Inventory (central inventory) directory.
- Create or reconfigure primary and secondary group memberships for the installation owner, if necessary, for the Oracle Inventory directory and the operating system privileges groups

- Set shell limits if necessary to required values.

Checking the Current Operating System Configuration

Instead of waiting for the installation to notify you of an incorrect configuration, you can use the command in this section to manually check the operating system configuration before installation. This helps you determine if additional time will be needed to update the operating system before starting the Oracle Grid Infrastructure installation.

To determine if the operating system requirements for Oracle Linux have been met:

1. To determine which distribution and version of Linux is installed, enter the one of the following commands at the operating system prompt as the `root` user:

   ```
   # cat /etc/oracle-release
   # cat /etc/redhat-release
   # lsb_release -id
   ```

2. To determine which chip architecture each server is using and which version of the software you should install, run the following command at the operating system prompt as the `root` user:

   ```
   # uname -m
   ```

   This command displays the processor type. For a 64-bit architecture, the output would be "x86_64".

3. To determine if the required errata level is installed, run the `uname` command as the `root` user, as shown in this example:

   ```
   # uname -r
   2.6.39-100.7.1.el6uek.x86_64
   ```

   Like most software, the Linux kernel is updated to fix bugs in the operating system. These kernel updates are referred to as erratum kernels or *errata levels*.

   The output in the previous example shows that the kernel version is 2.6.39, and the errata level (EL) is 100.7.1.e16uek. Review the required errata level for your distribution. If the errata level is below the required minimum errata level, then install the latest kernel update for your operating system. The kernel updates are available from your operating system vendor.

4. To ensure there are no operating system issues affecting installation, make sure you have installed all the operating system patch updates and packages that are listed in *Oracle Grid Infrastructure Installation Guide* for your platform. If you are using Oracle Linux, then you can determine if the required *packages*, or programs that perform specific functions or calculations, are installed by using the following command as the `root` user:

   ```
   # rpm -q package_name
   ```

   The variable *`package_name`* is the name of the package you are verifying, such as `setarch`. If a package is not installed, then install it from your Linux distribution media or download the required package version from your Linux vendor's website.

   You can also use either `up2date` or YUM (Yellow dog Updater Modified) to install packages and their dependencies on some Linux systems. YUM uses repositories to automatically locate and obtain the correct RPM packages for your system.

See Also:

- "About Installing Oracle RAC on Different Operating Systems"
- "Preparing the Server"
- "Preparing the Operating System and Software"
- "About Configuring the Software Owner's Shell Environment on Linux Systems"
- "About Performing Platform-Specific Configuration Tasks"
- *Oracle Grid Infrastructure Installation Guide* and the *Oracle Real Application Clusters Installation Guide* for your platform

Preparing the Server

After you have verified that your system meets the basic requirements for installing Oracle RAC, the next step is to configure the server in preparation for installation.

- About Operating System Users and Groups
- Configuring Secure Shell on Linux Systems
- About Configuring the Software Owner's Shell Environment on Linux Systems

See Also:

- "Preparing the Operating System and Software"
- "About Configuring Kernel Parameters"
- "About Configuring the Software Owner's Shell Environment on Linux Systems"
- "About Performing Platform-Specific Configuration Tasks"

About Operating System Users and Groups

Depending on whether this is the first time Oracle software is being installed on this server, you may have to create operating system groups and users.

The following topics discuss the types of users required for Oracle Clusterware and Oracle RAC:

- Required Operating System Users and Groups
- Separate Operating System Users and Groups for Oracle Software Installations on UNIX and Linux Systems
- Separate Operating System Users and Groups for Oracle Software Installations on Windows Systems
- Optional Operating System Users and Groups

See Also:

- "Preparing the Operating System and Software"
- "About Configuring the Software Owner's Shell Environment on Linux Systems"
- "Preparing the Server"

Required Operating System Users and Groups

To install Oracle Grid Infrastructure for a cluster and Oracle RAC, the operating system groups and users in the following table are required:

Operating System User or Group	Description
The Oracle Inventory group for all Oracle software installations (oinstall).	The Oracle Inventory group must be the primary group for Oracle software installation owners on Linux and UNIX platforms. Members of the Oracle Inventory group have access to the Oracle Inventory directory. This directory is the central inventory record of all Oracle software installations on a server and the installation logs and trace files from each installation. On Windows systems, this group is created and managed automatically.
An Oracle software owner, or installation user. This is the user account you use when installing the software.	If you want to use a single software owner for all installations, then typically the user name is oracle. If you plan to install Oracle Grid Infrastructure for a cluster and Oracle RAC using separate software owners to separate Oracle Grid Infrastructure for a cluster administrative privileges from Oracle Database administrative privileges, then typically you would use grid for Oracle Grid Infrastructure for a cluster owner and oracle for the Oracle RAC owner.
The OSDBA group (typically, dba) for Oracle Database authentication.	The OSDBA group is a system privileges group whose members are granted the SYSDBA privilege to administer Oracle Database and the SYSASM privilege to administer Oracle Clusterware and Oracle ASM. To provide fine-grained control of administrative privileges you can create multiple operating system groups as described in "Optional Operating System Users and Groups" on page 2-12.

Note: When installing Oracle RAC on a Microsoft Windows platform:

- OUI automatically creates the groups used for systsem authenticating access to the Oracle software.
- The user performing the installation must be an Administrator user.
- If you specify an Oracle Home user during installation, this user must be a domain user.

On Linux and UNIX systems, if you use one installation owner for both Oracle Grid Infrastructure for a cluster and Oracle RAC, then when you want to perform administration tasks, you must change the value for ORACLE_HOME environment variable to match the instance you want to administer (Oracle ASM, in the Grid home, or a database instance in the Oracle home). To change the ORACLE_HOME environment variable, use a command syntax similar to the following example, where /u01/app/12.1.0/grid is the Oracle Grid Infrastructure for a cluster home:

```
ORACLE_HOME=/u01/app/12.1.0/grid; export ORACLE_HOME
```

If you try to administer an instance using SQL*Plus, LSNRCTL, or ASMCMD while ORACLE_HOME is set to an incorrect binary path, then you will encounter errors. The Oracle home path does not affect SRVCTL commands.

On Windows systems, to perform administration tasks, you simply run the appropriate utility for either Oracle Grid Infrastructure or Oracle RAC and ORACLE_HOME is configured automatically.

See Also:

- "Preparing the Operating System and Software"
- "About Configuring the Software Owner's Shell Environment on Linux Systems"
- "Preparing the Server"

Separate Operating System Users and Groups for Oracle Software Installations on UNIX and Linux Systems

Instead of using a single operating system user as the owner of every Oracle software installation, you can use multiple users, each owning one or more Oracle software installations. A user created to own only the Oracle Grid Infrastructure for a cluster installation is called the `grid` user. This user owns both the Oracle Clusterware and Oracle Automatic Storage Management binaries. A user created to own either all Oracle software installations (including Oracle Grid Infrastructure for a cluster), or only Oracle Database software installations, is called the `oracle` user.

You can also use different users for each Oracle Database software installation. Additionally, you can specify a different OSDBA group for each Oracle Database software installation. By using different operating system groups for authenticating administrative access to each Oracle Database installation, users have `SYSDBA` privileges for the databases associated with their OSDBA group, rather than for all the databases on the system.

Members of the OSDBA group can also be granted the `SYSASM` system privilege, which gives them administrative access to Oracle ASM. As described in "Optional Operating System Users and Groups", you can configure a separate operating system group for Oracle ASM authentication to separate users with `SYSASM` access to the Oracle ASM instances from users with `SYSDBA` access to the database instances.

If you want to create separate Oracle software owners so you can use separate users and operating system privilege groups for the different Oracle software installations, then note that each of these users must have the Oracle central inventory group (`oinstall`) as their primary group. Members of this group have the required write privileges to the Oracle Inventory directory.

Note: The Oracle Grid Infrastructure for a cluster installation can be owned by only one user. You cannot have one user that owns the Oracle Clusterware installed files and a different user that owns the Oracle ASM installed files.

See Also:

- "About Operating System Users and Groups"
- "Optional Operating System Users and Groups"
- "About Configuring the Software Owner's Shell Environment on Linux Systems"

Separate Operating System Users and Groups for Oracle Software Installations on Windows Systems

You create at least one Oracle Installation user (an Administrator user who installs Oracle software) when you install Oracle Grid Infrastructure. You can use this Oracle

Installation user or a different Oracle Installation user when installing Oracle Database software.

You should create additional non-privileged Windows user accounts to be used as Oracle Home users during installation. An Oracle Home User is different from an Oracle Installation user. An Oracle Installation user is the user who needs administrative privileges to install Oracle products. An Oracle Home user is a low-privileged Windows User Account specified during installation that runs most of the Windows services required for the Oracle home. Different Oracle homes on a system can share the same Oracle Home user or use different Oracle Home users. You do not need to create any operating system groups because they are created and managed automatically by the Oracle software.

The Oracle Home user can be a Built-in Account or a Windows domain user. The Windows domain user should be a low-privileged, non-administrative account to ensure that the Oracle Database services running under Oracle Home User have only those privileges required to run Oracle products. Create separate Oracle Home users for each Oracle software product to have separate administrative controls for the different Oracle Homes.

If you use different Oracle Home users for each Oracle Database software installation, then you can use the `ORA_`*`HOMENAME`*`_DBA` group that is associated with each Oracle home to separate SYSDBA privileges for each installation. A member of an `ORA_`*`HOMENAME`*`_DBA` group can use operating system authentication to log in to only the Oracle Databases that run from the Oracle home with the name *`HOMENAME`*. Members of the `ORA_DBA` group can use operating system authentication to log in to any Oracle database with SYSDBA privileges.

You can also use the Oracle ASM access control feature to enable role separation for Oracle ASM management. In previous releases, this feature was disabled on Windows because all Oracle Database services ran as Local System. For Oracle ASM administration, the new groups `ORA_ASMADMIN`, `ORA_ASMDBA` and `ORA_ASMOPER` are automatically created and populated during Oracle Grid Infrastructure installation.

See Also:

- "About Operating System Users and Groups"
- "Optional Operating System Users and Groups"
- "About Configuring the Software Owner's Shell Environment on Linux Systems"

Optional Operating System Users and Groups

You can use additional users and groups to divide administrative access privileges to the Oracle Grid Infrastructure for a cluster installation from other administrative users and groups associated with other Oracle installations. Separating administrative access is implemented by specifying membership in different operating system groups, and separating installation privileges is implemented by using different installation owners for each Oracle installation.

The optional users and groups you can use are:

- The Oracle Automatic Storage Management Group, or OSASM group (typically `asmadmin` for Linux and `ORA_ASMADMIN` for Windows).
- The ASM Database Administrator group, or OSDBA for ASM group (typically `asmdba` for Linux and ORA_ASMDBA for Windows).

- The OSOPER for Oracle Database group (typically, oper on Linux and ORA_OPER on Windows).
- The OSOPER for Oracle ASM group (typically asmoper on Linux and ORA_ASMOPER on Windows).
- The OSBACKUPDBA group for Oracle Database.
- The OSDGDBA group for Oracle Data Guard.
- The OSKMDBA group for encryption key management.

> **See Also:**
>
> - "About Configuring the Software Owner's Shell Environment on Linux Systems"
> - "Preparing the Server"
> - *Oracle Grid Infrastructure Installation Guide* for your operating system for more information about configuring separation of privileges by role.

Configuring Operating System Users and Groups on Linux Systems

In this guide, a single software owner is used for all installations, named oracle. The oracle user belongs to the oinstall and dba operating system groups.

> **See Also:** *Oracle Grid Infrastructure Installation Guide* for your operating system for specific instructions

To create one software owner with all operating system-authenticated administration privileges:

1. Determine the groups that exist on your server by listing the contents of the /etc/group file.

   ```
   cat /etc/group
   ```

2. If this is the first time Oracle software has been installed on your server, and the Oracle Inventory group does not exist, then create the Oracle Inventory group (oinstall) with a group ID that is currently not in use on all the nodes in your cluster. Enter a command as the root user that is similar to the following:

   ```
   # /usr/sbin/groupadd -g 1000 oinstall
   ```

3. Create an OSDBA (dba) group with a group ID that is currently not in use on all the nodes in your cluster by entering a command as the root user that is similar to the following:

   ```
   # /usr/sbin/groupadd -g 1001 dba
   ```

4. If the user that owns the Oracle software (oracle) does not exist on your server, then you must create the user. Select a user ID (UID) that is currently not in use on all the nodes in your cluster. To determine which users have been created on your server, list the contents of the /etc/passwd file using the command:

   ```
   cat /etc/passwd
   ```

   The following command shows how to create the oracle user and the user's home directory (/home/oracle) with the default group as oinstall and the secondary group as dba, using a UID of 1100:

```
# useradd -u 1100 -g oinstall -G dba -d /home/oracle -r oracle
```

5. Set the password for the `oracle` account using the following command. Replace *`password`* with your own password.

   ```
   passwd oracle

   Changing password for user oracle.
   New UNIX password: password
   retype new UNIX password: password
   passwd:  all authentication tokens updated successfully.
   ```

6. Repeat Step 1 through Step 5 on each node in your cluster.

7. Verify that the attributes of the user `oracle` are identical on each node of your cluster:

   ```
   id oracle
   ```

   The command output should be similar to the following:

   ```
   uid=1100(oracle) gid=1000(oinstall) groups=1000(oinstall),1001(dba)
   ```

See Also:

- "Configuring Installation Directories and Shared Storage"
- "About Oracle Automatic Storage Management"

Configuring Secure Shell on Linux Systems

To install Oracle software, Secure Shell (SSH) connectivity must be set up between all cluster member nodes. OUI uses the `ssh` and `scp` commands during installation to run remote commands on and copy files to the other cluster nodes. You must configure SSH so that these commands do not prompt for a password. SSH is also used by the configuration assistants, Enterprise Manager, and when adding nodes to the cluster.

You can configure SSH from the Oracle Universal Installer (OUI) interface during installation for the user account running the installation. The automatic configuration creates passwordless SSH connectivity between all cluster member nodes.

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

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

- A given user has the same user name, user ID (UID), and password.
- A given user belongs to the same groups.
- A given group has the same group ID (GID).

See Also:

- "Configuring Operating System Users and Groups on Linux Systems" on page 2-13 for information about configuring user equivalency
- *Oracle Grid Infrastructure Installation Guide* for your platform for more information about manually configuring SSH

About Configuring the Software Owner's Shell Environment on Linux Systems

On Oracle Linux, you run Oracle Universal Installer (OUI) from the `oracle` account. OUI obtains information from the environment variables configured for the `oracle` user. Before running OUI, you must make the following changes to the shell startup file for the software owner of Oracle Grid Infrastructure for a cluster:

- Set the default file mode creation mask (`umask`) of the installation user (`oracle`) to 022 in the shell startup file. Setting the mask to 022 ensures that the user performing the software installation creates files with 644 permissions.
- Set `ulimit` settings for file descriptors (`nofile`) and processes (`nproc`) for the installation user (`oracle`).
- Set the software owner's environment variable `DISPLAY` environment variables in preparation for the Oracle Grid Infrastructure for a cluster installation.
- Remove any lines in the file that set values for `ORACLE_SID`, `ORACLE_HOME`, or `ORACLE_BASE` environment variables.

After you have saved your changes, run the shell startup script to configure the environment.

Also, if the `/tmp` directory has less than 1 GB of available disk space, but you have identified a different, non-shared file system that has at least 1 GB of available space, then you can set the `TEMP` and `TMPDIR` environment variables to specify the alternate temporary directory on this file system.

To review your current environment settings, use the `env | more` command as the `oracle` user.

Refer to the *Oracle Grid Infrastructure Installation Guide* for your platform for more information on how to configure the Oracle software owner environment before installation.

Note: Remove any `stty` commands from hidden files (such as logon or profile scripts) before you start the installation. On Linux systems, if there are any such files that contain `stty` commands, then when these files are loaded by the remote shell during installation, OUI indicates an error and stops the installation.

See Also:

- "About Operating System Users and Groups"
- "Preparing the Operating System and Software"
- "Configuring Installation Directories and Shared Storage"
- "About Setting the Time on All Nodes"
- "About Performing Platform-Specific Configuration Tasks"

Configuring the Network

Oracle Database 12*c* Release 1 (12.1) clients connect to the database using a single client access name (SCAN). The SCAN and its associated IP addresses provide a stable name for clients to use for connections, independent of the nodes that form the cluster. A SCAN works by being able to resolve to multiple IP addresses referencing multiple listeners in the cluster that handle public client connections. The installation process creates these listeners, called SCAN listeners.

To configure the network in preparation for installing Oracle Grid Infrastructure for a cluster:

1. Determine your cluster name. The cluster name should satisfy the following conditions:

   - The cluster name is at least one character long and less than 15 characters long.
   - The cluster name must consist of the same character set used for host names: single-byte alphanumeric characters (a to z, A to Z, and 0 to 9) and hyphens (-).
   - The cluster name is globally unique throughout your host domain
   - If you use third-party vendor clusterware, then Oracle recommends that you use the vendor cluster name.

2. Determine the public host name and virtual host name for each node in the cluster.

   - For the public host name, use the primary host name of each node. In other words, use the name displayed by the `hostname` command. This host name can be either the permanent or the virtual host name, for example: `racnode1`.
   - Determine a virtual host name for each node. A virtual host name is a public node name that Oracle Clusterware uses to reroute client requests sent to the node if the node is down. Oracle recommends that you provide a name in the format `<`*`public hostname`*`>-vip`, for example: `racnode1-vip`.

3. Identify the interface names and associated IP addresses for all network adapters by executing the following command on each node:

   ```
   # /sbin/ifconfig
   ```

   From the output, identify the interface name (such as `eth0`) and IP address for each network adapter you specify as a public network interface.

   If your operating system supports a graphic user interface (GUI) for modifying the system configuration, then you can also use the following command to start a GUI that you can use to configure the network adapters and `/etc/hosts` file:

   ```
   /usr/bin/system-config-network &
   ```

   > **Note:** When you install Oracle Clusterware and Oracle RAC, you are asked to provide the interface name and IP address for each network adapter.

4. On each node in the cluster, assign a public IP address with an associated network name to one network adapter. The public name for each node should be registered with your domain name system (DNS). IP addresses on the subnet you identify as private are assigned as private IP addresses for cluster member nodes. You do not have to configure these addresses manually in the `/etc/hosts` file.

You can test whether an interconnect interface is reachable using a `ping` command.

5. On each node in the cluster, configure a third IP address that serves as a virtual IP address. Use an IP address that meets the following requirements:

   - The virtual IP address and the network name must *not* be currently in use.
   - The virtual IP address must be on the *same* subnet as your public IP address.

   The virtual host name for each node should be registered with your DNS.

6. Define a SCAN that resolves to three IP addresses in your DNS.

7. When you complete the network configuration, the IP address and network interface configuration should be similar to what is shown in Table 2–1 (your node names and IP addresses might be different):

Table 2–1 Manual Network Configuration Example

Identity	Home Node	Host Node	Given Name	Type	Address	Address Assigned By	Resolved By
Node 1 Public	Node 1	`racnode1`	`racnode1`[1]	Public	192.0.2.101	Fixed	DNS
Node 1 VIP	Node 1	Selected by Oracle Clusterware	`racnode1-vip`	Virtual	192.0.2.104	Fixed	DNS, hosts file
Node 1 Private	Node 1	`racnode1`	`racnode1-priv`	Private	192.168.0.1	Fixed	DNS, hosts file, or none
Node 2 Public	Node 2	`racnode2`	`racnode2`[1]	Public	192.0.2.102	Fixed	DNS
Node 2 VIP	Node 2	Selected by Oracle Clusterware	`racnode2-vip`	Virtual	192.0.2.105	Fixed	DNS, hosts file
Node 2 Private	Node 2	`racnode2`	`racnode2-priv`	Private	192.168.0.2	Fixed	DNS, hosts file, or none
SCAN VIP 1	none	Selected by Oracle Clusterware	`docrac-scan`	virtual	192.0.2.201	Fixed	DNS
SCAN VIP 2	none	Selected by Oracle Clusterware	`docrac-scan`	virtual	192.0.2.202	Fixed	DNS
SCAN VIP 3	none	Selected by Oracle Clusterware	`docrac-scan`	virtual	192.0.2.203	Fixed	DNS

[1] Node host names may resolve to multiple addresses.

Even if you are using a DNS, Oracle recommends that you add lines to the `/etc/hosts` file on each node, specifying the public IP addresses. Configure the `/etc/hosts` file so that it is similar to the following example:

```
#eth0 - PUBLIC
192.0.2.100       racnode1.example.com       racnode1
192.0.2.101       racnode2.example.com       racnode2
```

Note: Make a note of the addresses you configured in the /etc/hosts file or registered with DNS. When you install Oracle Grid Infrastructure for a cluster and Oracle RAC, you are prompted for the public, virtual IP, and SCAN addresses.

Oracle strongly recommends that you do not configure SCAN VIP addresses in the hosts file. Use DNS resolution for SCAN VIPs.

The fully qualified SCAN for the cluster defaults to *cluster_name*-scan.*GNS_subdomain_name*, for example docrac-scan.grid.example.com. The short SCAN for the cluster is docrac-scan. Without GNS, and configuring SCAN in DNS directly, then the fully qualified SCAN would be similar to docrac-scan.example.com. You can use any name for the SCAN, if it is unique within your network and conforms to the RFC 952 standard.

8. If you configured the IP addresses in a DNS server, then, as the root user, change the hosts search order in /etc/nsswitch.conf on all nodes as shown here:

   Old:

   ```
   hosts:  files nis dns
   ```

   New:

   ```
   hosts:  dns files nis
   ```

9. After modifying the nsswitch.conf file, restart the nscd daemon on each node using the following command:

   ```
   # /sbin/service nscd restart
   ```

After you have completed the installation process, configure clients to use the SCAN to access the cluster. Using the previous example, the clients would use docrac-scan to connect to the cluster.

See Also:

- "About Network Hardware Requirements"
- *Oracle Grid Infrastructure Installation Guide* for your platform

Verifying the Network Configuration

After you have configured the network, perform verification tests to make sure it is configured properly. If there are problems with the network connection between nodes in the cluster, then the Oracle Clusterware installation fails.

To verify the network configuration on a two-node cluster that is running Oracle Linux:

1. As the root user, verify the configuration of the public and private networks. Verify that the interfaces are configured on the same network (either private or public) on all nodes in your cluster.

   In this example, eth0 is used for the public network and eth1 is used for the private network on each node.

   ```
   # /sbin/ifconfig

   eth0      Link encap:Ethernet  HWaddr 00:0E:0C:08:67:A9
   ```

```
          inet addr: 192.0.2.100   Bcast:192.0.2.255   Mask:255.255.240.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:270332689 errors:0 dropped:0 overruns:0 frame:0
          TX packets:112346591 errors:2 dropped:0 overruns:0 carrier:2
          collisions:202 txqueuelen:1000
          RX bytes:622032739 (593.2 MB)  TX bytes:2846589958 (2714.7 MB)
          Base address:0x2840 Memory:fe7e0000-fe800000

eth1      Link encap:Ethernet  HWaddr 00:04:23:A6:CD:59
          inet addr: 10.10.10.11   Bcast: 10.10.10.255   Mask:255.255.240.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21567028 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15259945 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4091201649 (3901.6 MB)  TX bytes:377502797 (360.0 MB)
          Base address:0x2800 Memory:fe880000-fe8a0000
```

2. As the root user, verify the network configuration by using the ping command to test the connection from each node in your cluster to all the other nodes. For example, as the root user, you might run the following commands on each node:

   ```
   # ping -c 3 racnode1.example.com
   # ping -c 3 racnode1

   # ping -c 3 racnode2.example.com
   # ping -c 3 racnode2
   ```

   You should not get a response from the nodes using the ping command for the virtual IPs (racnode1-vip, racnode2-vip) or the SCAN IPs until after Oracle Clusterware is installed and running. If the ping commands for the public addresses fail, then resolve the issue before you proceed.

3. Ensure that you can access the default gateway with a ping command. To identify the default gateway, use the route command, as described in the Oracle Linux Help utility.

   See Also:

   - "Checking the Settings for the Interconnect"
   - "Configuring the Network"
   - "About Network Hardware Requirements"

Preparing the Operating System and Software

When you install the Oracle software on your server, Oracle Universal Installer expects the operating system to have specific packages and software applications installed.

This section covers the following topics:

- About Setting the Time on All Nodes
- About Configuring Kernel Parameters
- About Performing Platform-Specific Configuration Tasks

See Also:

- "Preparing the Server"
- "Verifying Operating System and Software Requirements"

About Setting the Time on All Nodes

Before starting the installation, ensure that the date and time settings on all the cluster nodes are set as closely as possible to the same date and time. A cluster time synchronization mechanism ensures that the internal clocks of all the cluster members are synchronized. For Oracle RAC on Linux, you can use either the Network Time Protocol (NTP) or the Oracle Cluster Time Synchronization Service.

NTP is a protocol designed to synchronize the clocks of servers connected by a network. When using NTP, each server on the network runs client software to periodically make timing requests to one or more servers, referred to as reference NTP servers. The information returned by the timing request is used to adjust the server's clock. All the nodes in your cluster should use the same reference NTP server.

Note: If you use NTP on Linux or UNIX platforms, then you must configure it using the `-x` flag. See the *Oracle Grid Infrastructure Installation Guide* for your platform for details on how to configure NTP.

If you do not configure NTP, then Oracle configures and uses the Cluster Time Synchronization Service (CTSS). CTSS can also be used to synchronize the internal clocks of all the members in the cluster. CTSS keeps the member nodes of the cluster synchronized. CTSS designates the first node in the cluster as the master and then synchronizes all other nodes in the cluster to have the same time as the master node. CTSS does not use any external clock for synchronization.

Note: Using NTP or CTSS does not protect your system against human error resulting from a change in the system time for a node.

See Also:

- "Preparing the Server"
- "Preparing the Operating System and Software"

About Configuring Kernel Parameters

OUI checks the current settings for various kernel parameters to ensure they meet the minimum requirements for deploying Oracle RAC. For production database systems, Oracle recommends that you tune the settings to optimize the performance of your particular system.

Note: If you find parameter settings or shell limit values on your system that are greater than the values mentioned by OUI, then *do not* modify the parameter setting.

See Also:

- "Preparing the Server"
- "Preparing the Operating System and Software"
- *Oracle Grid Infrastructure Installation Guide* for your platform

About Performing Platform-Specific Configuration Tasks

You may be required to perform special configuration steps that are specific to the operating system on which you are installing Oracle RAC, or for the components used with your cluster. The following list provides examples of operating-specific installation tasks:

- Configure the use of Huge Pages on SUSE Linux, Red Hat Enterprise Linux, or Oracle Linux.
- Set shell limits for the `oracle` user on Red Hat Linux or Oracle Linux systems to increase the number of files and processes available to Oracle Clusterware and Oracle RAC.
- Create X library symbolic links on HP-UX.
- Configure network tuning parameters on AIX Based Systems.

See Also:

- "Preparing the Server"
- "Preparing the Operating System and Software"
- "About Installing Oracle RAC on Different Operating Systems"
- *Oracle Grid Infrastructure Installation Guide* for your platform

Configuring Installation Directories and Shared Storage

Oracle RAC requires access to a shared file system for storing Oracle Clusterware files. You must also determine where the Oracle software and database files will be installed.

This section describes the storage configuration tasks that you must complete before you start Oracle Universal Installer. It includes information about the following topics:

- Locating the Oracle Inventory Directory
- Creating the Oracle Grid Infrastructure for a Cluster Home Directory
- Creating the Oracle Base Directory
- About the Oracle Home Directory
- Configuring Shared Storage
- Configuring Files on an NAS Device for Use with Oracle ASM
- Using ASMLIB to Mark the Shared Disks as Candidate Disks

Note: Additional configuration is required to use the Oracle Automatic Storage Management library Driver (ASMLIB) with third party vendor multipath disks. Refer to *Oracle Grid Infrastructure Installation Guide* for your operating system for more details about these requirements.

See Also:

- "Verifying System Requirements"
- "About Operating System Users and Groups"
- "About Hardware Requirements"
- The "Configuring Storage.." chapter in *Oracle Grid Infrastructure Installation Guide* for your platform

Locating the Oracle Inventory Directory

The Oracle Inventory (`oraInventory`) directory is the central inventory record of all Oracle software installations on a server. The `oraInventory` directory contains the following:

- A registry of the Oracle home directories (Oracle Grid Infrastructure for a cluster and Oracle Database) on the system
- Installation logs and trace files from installations of Oracle software. These files are also copied to the respective Oracle homes for future reference

OUI creates the `oraInventory` directory during installation. By default, the Oracle Inventory directory is not installed under the Oracle Base directory. This is because all Oracle software installations share a common Oracle Inventory, so there is only one Oracle Inventory for all users, whereas each user that performs a software installation can use a separate Oracle base directory.

If you have an existing Oracle Inventory, then ensure that you use the same Oracle Inventory for all Oracle software installations, and ensure that all Oracle software users you intend to use for installation have permissions to write to this directory.

To determine if you have an Oracle central inventory directory (oraInventory) on your system:

1. Run the following command to check for an existing Oracle Inventory directory:

   ```
   # more /etc/oraInst.loc
   ```

2. If the `oraInst.loc` file exists, then the output from this command is similar to the following:

   ```
   inventory_loc=/u01/app/oraInventory
   inst_group=oinstall
   ```

   In the previous output example:

   - The `inventory_loc` group shows the location of the Oracle Inventory.
   - The `inst_group` parameter shows the name of the Oracle Inventory group (in this example, `oinstall`).

3. If the Oracle Inventory directory does not exist, then the command returns an error message indicating that the file or directory does not exist.

   If the Oracle Inventory directory does not exist, then you do not have to create one before installing the Oracle software.

See Also:

- "Verifying System Requirements"
- "About Operating System Users and Groups"
- "About Hardware Requirements"
- *Oracle Grid Infrastructure Installation Guide* for your platform

Creating the Oracle Grid Infrastructure for a Cluster Home Directory

During installation, you are prompted to provide a path to a home directory in which to place the software for Oracle Grid Infrastructure for a cluster. OUI installs Oracle Clusterware and Oracle ASM into a directory referred to as *Grid_home*. Ensure that the directory path you provide meets the following requirements:

- It should be created in a path outside existing Oracle homes.
- It should not be located in a user home directory.
- It should be created either as a subdirectory in a path where all files can be owned by `root`, or in a unique path.
- If you create the path before installation, then it should be owned by the installation owner of Oracle Grid Infrastructure for a cluster (`oracle` or `grid`), and set to 775 permissions.

Before you start the installation, you must have sufficient disk space on a file system for the Oracle Grid Infrastructure for a cluster directory. The file system that you use for the Grid home directory must have at least 4.5 GB of available disk space.

The path to the Grid home directory must be the same on all nodes. As the `root` user, you should create a path compliant with Oracle Optimal Flexible Architecture (OFA) guidelines, so that OUI can select that directory during installation.

To create the Grid home directory:

Log in to the operating system as the `root` user, then enter the following commands, where `oracle` is the name of the user that will install the Oracle Grid Infrastructure software:

```
# mkdir -p /u01/app/12.1.0/grid
# chown -R oracle:oinstall /u01/app/12.1.0/grid
```

Note: On Linux and UNIX systems, you must ensure the Grid home directory is not a subdirectory of the Oracle base directory. Installing Oracle Clusterware in an Oracle base directory causes installation errors.

See Also:

- "About Hardware Requirements"
- "Configuring Shared Storage"

Creating the Oracle Base Directory

Oracle Universal Installer (OUI) creates the Oracle base directory for you in the location you specify. This directory is owned by the user performing the installation.

The Oracle base directory (`ORACLE_BASE`) helps to facilitate the organization of Oracle installations, and helps to ensure that installations of multiple databases maintain an Optimal Flexible Architecture (OFA) configuration. OFA guidelines recommend that you use a path similar to the following for the Oracle base directory:

```
/mount_point/app/user
```

In the preceding path example, the variable *`mount_point`* is the mount point directory for the file system where you intend to install the Oracle software and *`user`* is the Oracle software owner (typically `oracle`). For OUI to recognize the path as an Oracle software path, it must be in the form `u0[1-9]/app`, for example, `/u01/app`.

The path to the Oracle base directory must be the same on all nodes. The permissions on the Oracle base directory should be at least 750.

Example: Creating an Oracle Base Directory

Before installing any Oracle software, you must configure an Oracle base directory, OUI uses the Oracle base directory to determine the location of the Oracle Inventory directory, and also for installing Oracle RAC.

Assume you have determined that the file system mounted as `/u01` has sufficient room for both the Oracle Grid Infrastructure for a cluster and Oracle RAC software. You have decided to make the `/u01/app/oracle/` directory the Oracle base directory. The user installing all the Oracle software is the `oracle` user.

To create the Oracle base directory:

1. Log in as the `root` user.
2. Use the `mkdir` command to create the path to the Oracle base directory.

   ```
   # mkdir -p /u01/app/oracle/
   ```

3. Change the ownership of the Oracle base path to the Oracle software owner, `oracle`.

   ```
   # chown -R oracle:oinstall /u01/app/oracle/
   ```

4. Change the permissions on the Oracle base directory to 775.

   ```
   # chmod -R 775 /u01/app/oracle/
   ```

See Also:

- "About Hardware Requirements"
- "Configuring Shared Storage"

About the Oracle Home Directory

The Oracle home directory is the location in which the Oracle RAC software is installed. You can use an Oracle home directory created in the local file system, for example, `/u01/app/oracle/product/12.1.0/dbhome_1`. The same directory must exist on every node in the cluster. You do not have to create these directories before installation. By default, the installer suggests a subdirectory of the Oracle base directory for the Oracle home.

You can also use a shared Oracle home. The location of the shared Oracle home can be on network storage, or a supported cluster file system such as Oracle Automatic Storage Management Cluster File System (Oracle ACFS). For more information about Oracle ACFS, see *Oracle Automatic Storage Management Administrator's Guide*.

If you use the local file system for the Oracle home directory, and you want to install a different release of Oracle RAC or Oracle Database on the same server, then you must use a separate Oracle home directory for each software installation. Multiple releases of the same product or different products can run from different Oracle homes concurrently. Products installed in one Oracle home do not conflict or interact with products installed in another Oracle home.

Using different Oracle homes for your installed software enables you to perform maintenance operations on the Oracle software in one home without effecting the software in another Oracle home. However, it also increases your software maintenance costs, because each Oracle home must be upgraded or patched separately.

See Also:

- "Verifying System Requirements"
- "About Operating System Users and Groups"
- "About Hardware Requirements"
- *Oracle Grid Infrastructure Installation Guide* for your platform

Configuring Shared Storage

Each node in a cluster requires external shared disks for storing the Oracle Clusterware (Oracle Cluster Registry and voting disk) files, and Oracle Database files. The supported types of shared storage depend upon the platform you are using, for example:

- Oracle Automatic Storage Management (strongly recommended)
- A supported cluster file system, such as OCFS2 for Linux, or General Parallel File System (GPFS) on IBM platforms
- Network file system (NFS), which is not supported on Linux on POWER or on IBM zSeries Based Linux
- (Upgrades only) Shared disk partitions consisting of block devices. Block devices are disk partitions that are not mounted using the Linux file system. Oracle Clusterware and Oracle RAC write to these partitions directly.

> **Note:** You cannot use OUI to install Oracle Clusterware files on block or raw devices. You cannot put Oracle Clusterware binaries and files on Oracle Automatic Storage Management Cluster File System (Oracle ACFS).

If you decide to use OCFS2 to store the Oracle Clusterware files, then you must use the proper version of OCFS2 for your operating system version. OCFS2 works with Oracle Linux and Red Hat Linux kernel version 2.6

For all installations, you must choose the storage option to use for Oracle Clusterware files and Oracle Database files. The examples in this guide use Oracle ASM to store the Oracle Clusterware and Oracle Database files. The Oracle Grid Infrastructure for a cluster and Oracle RAC software is installed on disks local to each node, not on a shared file system.

This guide discusses two different methods of configuring the shared disks for use with Oracle ASM:

- Configuring Files on an NAS Device for Use with Oracle ASM
- Using ASMLIB to Mark the Shared Disks as Candidate Disks

Note: For the most up-to-date information about supported storage options for Oracle RAC installations, refer to the Certifications pages on My Oracle Support at:

`https://support.oracle.com`

See Also:

- *Oracle Grid Infrastructure Installation Guide* for your platform if you are using a cluster file system or NFS
- "Configuring Installation Directories and Shared Storage"
- "About Hardware Requirements"

Configuring Files on an NAS Device for Use with Oracle ASM

To use an NFS file system, it must be on a certified NAS device. If you have a certified network attached storage (NAS) device, then you can create zero-padded files in an NFS mounted directory and use those files as disk devices in an Oracle ASM disk group.

To ensure high availability of Oracle Clusterware files on Oracle ASM, you must have at least 2 GB of disk space for Oracle Clusterware files in three separate failure groups, with at least three physical disks. Each disk must have at least 1 GB of capacity to ensure that there is sufficient space to create Oracle Clusterware files.

Use the following guidelines when identifying appropriate disk devices:

- All of the devices in an Oracle Automatic Storage Management disk group should be the same size and have the same performance characteristics.
- A disk group should not contain multiple partitions on a single physical disk device.
- Using logical volumes as a device in an Oracle Automatic Storage Management disk group is not supported with Oracle RAC.
- The user account with which you perform the installation (`oracle`) must have write permissions to create the files in the path that you specify.

To configure NAS device files for creating disk groups:

1. Add and configure access to the disks to the NAS device. Make sure each cluster node has been granted access to all the disks used by Oracle Grid Infrastructure for a cluster and Oracle Database software.

   Refer to your NAS device documentation for more information about completing this step.

2. On a cluster node, log in as the `root` user (or use `sudo` for the following steps).

3. Configure access to the disks on the NAS devices. The process for completing this step can vary depending on the type of disks and the type of NAS service.

   One example of the configuration process is shown here. The first step is to create a mount point directory on the local system:

   ```
   # mkdir -p /mnt/oracleasm
   ```

4. To ensure that the NFS file system is mounted when the system restarts, add an entry for the file system in the mount file `/etc/fstab`.

   For more information about editing the mount file for the operating system, refer to the Linux man pages. For more information about recommended mount options, refer to *Oracle Grid Infrastructure Installation Guide* for your platform.

5. Enter a command similar to the following to mount the NFS file system on the local system, where *`host`* is the host name or IP address of the file server, and *`pathname`* is the location of the storage within NFS (for example, `/public`):

```
# mount <host>:<pathname> /mnt/oracleasm
```

6. Choose a name for the disk group to create, for example, `nfsdg`.

7. Create a directory for the files on the NFS file system, using the disk group name as the directory name, for example:

```
# mkdir /mnt/oracleasm/nfsdg
```

8. Use commands similar to the following to create the required number of zero-padded files in this directory:

```
# dd if=/dev/zero of=/mnt/oracleasm/nfsdg/disk1 bs=1024k count=1000
```

   This example creates a 1 GB file named `disk1` on the NFS file system. You must create one, two, or three files respectively to create an external, normal, or high redundancy disk group.

9. Enter the following commands to change the owner, group, and permissions on the directory and files that you created:

```
# chown -R oracle:dba /mnt/oracleasm
# chmod -R 660 /mnt/oracleasm
```

10. When installing Oracle RAC, if you choose to create an Oracle ASM disk group, then you must change the disk discovery path to specify a regular expression that matches the file names you created, for example, `/mnt/oracleasm/nfsdg/*`.

See Also:

- "Configuring Shared Storage"
- "About Hardware Requirements"

Using ASMLIB to Mark the Shared Disks as Candidate Disks

Another option for configuring shared disks is to use the ASMLIB utility. If you configure a shared disk to be mounted automatically when the server restarts, then, unless you have configured special files for device persistence, a disk that appeared as `/dev/sdg` before the system shutdown can appear as `/dev/sdh` after the system is restarted.

If you use ASMLIB to configure the shared disks, then when you restart the node:

- The disk device names do not change
- The ownership and group membership for these disk devices remains the same
- You can copy the disk configuration implemented by Oracle ASM to other nodes in the cluster by running a simple command

Note: If you followed the instructions in "Configuring Files on an NAS Device for Use with Oracle ASM" on page 2-26 to configure your shared storage, then you do not have to perform the tasks in this section.

The following topics describe how to install and configure ASMLib, and how to use ASMLib to configure your shared disk devices:

- Installing ASMLIB
- Configuring ASMLIB
- Using ASMLIB to Create Oracle ASM Disks

See Also: If you choose not to use ASMLIB, then see "Configuring Disk Device Persistence" on page 2-30

Installing ASMLIB

The ASMLIB software for various Linux platforms is available from the Oracle Technology Network. Select the link for your platform on the ASMLIB download page at:

`http://www.oracle.com/technetwork/topics/linux/asmlib/index-101839.html`

You should see four to six packages for your Linux platform. The `oracleasmlib` package provides the actual Oracle ASM library. The `oracleasm-support` package provides the utilities used to get the Oracle ASM driver up and running. Both of these packages must be installed.

The remaining packages provide the kernel driver for the Oracle ASM library. Each package provides the driver for a different kernel. You must install the appropriate package for the kernel you run. Use the `uname -r` command to determine the version of the kernel on your server. The `oracleasm` kernel driver package has that version string in its name. For example, if you run Red Hat Enterprise Linux 4 AS, and the kernel you are using is the 2.6.9-55.0.12.ELsmp kernel, then you would choose the `oracleasm-2.6.9-55.0.12.ELsmp-2.0.3-1.x86_64.rpm` package.

Note: The Oracle ASMLIB kernel driver (`oracleasm`) is included in the Unbreakable Enterprise Kernel for Oracle Linux. You do not have to installany driver packages when using this kernel. You must still install the `oracleasm-support` and `oracleasmlib` packages.

To install the ASMLIB software packages:

1. Download the ASMLIB packages to each node in your cluster.
2. Change to the directory where the package files were downloaded.
3. As the `root` user, use the `rpm` command to install the packages. For example:

```
# rpm -Uvh oracleasm-support-2.1.3-1.el4.x86_64.rpm
# rpm -Uvh oracleasmlib-2.0.4-1.el4.x86_64.rpm
# rpm -Uvh oracleasm-2.6.9-55.0.12.ELsmp-2.0.3-1.x86_64.rpm
```

   After you have completed these commands, ASMLIB is installed on the system.

4. Repeat steps 2 and 3 on each node in your cluster.

See Also:

- "Using ASMLIB to Mark the Shared Disks as Candidate Disks" on page 2-27
- "Configuring Shared Storage"
- "About Hardware Requirements"

Configuring ASMLIB

After the ASMLIB software is installed, a few steps have to be taken by the system administrator to make the Oracle ASM driver available. The Oracle ASM driver must be loaded, and the driver file system must be mounted. This is taken care of by the initialization script, `/usr/sbin/oracleasm`.

To configure the ASMLIB software after installation:

1. As the `root` user, run the following command:

   ```
   # /usr/sbin/oracleasm configure -I
   ```

   The script prompts you for the default user and group to own the Oracle ASM driver access point. Specify the Oracle Database software owner (`oracle`) and the OSDBA group (`dba`).

   The script also prompts you to specify whether you want to start the ASMLIB driver when the node is started and whether you want to scan for presence of any Oracle Automatic Storage Management disks when the node is started. Answer yes for both of these questions.

2. Repeat step 1 on each node in your cluster.

See Also:

- "Using ASMLIB to Mark the Shared Disks as Candidate Disks" on page 2-27
- "Configuring Shared Storage"
- "About Hardware Requirements"

Using ASMLIB to Create Oracle ASM Disks

Every disk that is used in an Oracle ASM disk group must be accessible on each node. After you make the physical disk available to each node, you can then mark the disk device as an Oracle ASM disk. The `/usr/sbin/oracleasm` script is used for this task.

If the target disk device supports partitioning, then you must first create a single partition that encompasses the entire disk. If the target disk device does not support partitioning, then you do not have to create a partition on the disk.

To create Oracle ASM disks using ASMLIB:

1. As the `root` user, use `oracleasm` to create Oracle ASM disks using the following syntax:

   ```
   # /usr/sbin/oracleasm createdisk disk_name device_partition_name
   ```

   In this command, *`disk_name`* is the name you choose for the Oracle ASM disk. The name you choose must contain only ASCII capital letters, numbers, or underscores, and the disk name must start with a letter, for example, DISK1 or

VOL1, or RAC_FILE1. The name of the disk partition to mark as an Oracle ASM disk is the *device_partition_name*. For example:

```
# /usr/sbin/oracleasm createdisk DISK1 /dev/sdb1
```

If you must unmark a disk that was used in a `createdisk` command, then you can use the following syntax:

```
# /usr/sbin/oracleasm deletedisk disk_name
```

2. Repeat step 1 for each disk that Oracle ASM uses.
3. After you have created all the Oracle ASM disks for your cluster, use the `listdisks` command to verify their availability:

```
# /usr/sbin/oracleasm listdisks
DISK1
DISK2
DISK3
```

4. On all the other nodes in the cluster, use the `scandisks` command to view the newly created Oracle ASM disks. You do not have to create the Oracle ASM disks on each node, only on one node in the cluster.

```
# /usr/sbin/oracleasm scandisks
Scanning system for ASM disks                              [  OK  ]
```

5. After scanning for Oracle ASM disks, display the available Oracle ASM disks on each node to verify their availability:

```
# /usr/sbin/oracleasm listdisks
DISK1
DISK2
DISK3
```

Note: At this point, you should restart each node on which you are installing Oracle Grid Infrastructure for a cluster. After the node has restarted, view the configured shared storage on each node. This helps to ensure that the system configuration is complete and persists across node shutdowns.

See Also:

- "Using ASMLIB to Mark the Shared Disks as Candidate Disks" on page 2-27
- "Configuring Shared Storage"
- "About Hardware Requirements"

Configuring Disk Device Persistence

By default, the Linux 2.6 kernel device file naming scheme `udev` dynamically creates device file names when the server is started, and assigns ownership of them to `root`. If `udev` applies default settings, then it changes device file names and owners for voting disks or Oracle Cluster Registry partitions, corrupting them when the server is restarted. For example, a voting disk on a device named `/dev/sdd` owned by the user `grid` may be on a device named `/dev/sdf` owned by `root` after restarting the server.

If you use ASMLIB, then you do not have to ensure permissions and device path persistency in `udev`. If you do not use ASMLIB, then you must create a custom rules file for the shared disks mounted on each node. When `udev` is started, it sequentially carries out rules (configuration directives) defined in rules files. These files are in the path `/etc/udev/rules.d/`. Rules files are read in lexical order. For example, rules in the file `10-wacom.rules` are parsed and carried out before rules in the rules file `90-ib.rules`.

Where rules files describe the same devices, on Asianux, Red Hat, and Oracle Linux, the last file read is the one that is applied. On SUSE 2.6 kernels, the first file read is the one that is applied.

To configure a rules file for disk devices, see the chapter on configuring storage in *Oracle Grid Infrastructure Installation Guide* for your platform.

See Also:

- "Configuring Shared Storage" on page 2-25
- *Oracle Grid Infrastructure Installation Guide* for your platform

3 Installing Oracle Grid Infrastructure and Oracle Real Application Clusters

Before installing Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node using Oracle Universal Installer (OUI), you must first install the Oracle Grid Infrastructure for a cluster software, which consists of Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM). After Oracle Clusterware and Oracle ASM are operational, you can use OUI to install the Oracle Database software with the Oracle RAC components.

- Using Oracle Enterprise Manager Cloud Control for Provisioning Operations
- Preparing to Install the Oracle Software
- Verifying My Oracle Support Credentials
- Installing Oracle Grid Infrastructure for a Cluster
- Installing the Oracle Database Software and Creating a Database
- Performing Postinstallation Tasks
- About Converting an Oracle Database to an Oracle RAC Database

See Also:

- "About Hardware Requirements"
- "Configuring Installation Directories and Shared Storage"
- "Verifying Operating System and Software Requirements"
- "About Operating System Users and Groups"

Using Oracle Enterprise Manager Cloud Control for Provisioning Operations

As part of the database provisioning solution, Oracle Enterprise Manager Cloud Control enables you to provision Oracle Databases (also known as single-instance databases) and Oracle Real Application Clusters databases, extend or delete Oracle Real Application Clusters nodes, provision Oracle Real Application Clusters One node databases. Oracle Enterprise Manager Cloud Control also helps in setting up the environment for provisioning and does various pre-checks on the host to verify provisioning is successful. Before you can use Oracle Enterprise Manager Cloud Control to provision your Oracle software, you must perform the following system configuration tasks:

- Discover all host by installing the Enterprise Manager Agent on all nodes.

- Configure named and preferred credentials in Enterprise Manager.
- Configure a software library and setup roles.
- Download software from Oracle Techology Network to create installation media components in Enterprise Manager. Optionally, you can create a Provisioning Profile (similar to a gold image) from an existing system.

Details on how to perform these tasks, and how to provision your Oracle Grid Infrastructure and Oracle RAC software using Oracle Enterprise Manager Cloud Control are available in *Oracle Enterprise Manager Lifecycle Management Administrator's Guide*.

The rest of this chapter describes how to install without using Cloud Control.

Preparing to Install the Oracle Software

Oracle Clusterware is installed as part of Oracle Grid Infrastructure for a cluster. OUI installs Oracle Clusterware into a directory structure called the *Grid_home*. This home is separate from the home directories of other Oracle software products installed on the same server. Because Oracle Clusterware works closely with the operating system, you must have system administrator privileges for some installation tasks. In addition, some Oracle Clusterware processes must run as the special operating system user, root. OUI also installs Oracle Automatic Storage Management (Oracle ASM) in the Grid home directory.

You install the Oracle RAC software from the Oracle Database 12*c* installation media. By default, the standard Oracle Database 12*c* software installation process installs the Oracle RAC option when OUI recognizes that you are performing the installation on a cluster. OUI installs Oracle RAC into a directory structure called the *Oracle_home*. This home is separate from the home directories of other Oracle software products installed on the same server.

To prepare to install the Oracle software:

1. If you have the Oracle software on CDs or DVDs, then insert the disc for Oracle Grid Infrastructure for a cluster into a disk drive on your computer. Make sure the disk drive has been mounted at the operating system level. You must change discs when installing the Oracle Database software.

   If you do not have installation disks, but are instead installing from ZIP files, then continue to Step 2.

2. If the installation software is in one or more ZIP files, then create a staging directory on one node, for example, racnode1, to store the unzipped files, as shown here:

   ```
   mkdir -p /stage/oracle/12.1.0
   ```

3. Copy the ZIP files to this staging directory. For example, if the files were downloaded to a directory named /home/user1, and the ZIP file is named 12100_linux_db.zip, then use the following command to move the ZIP file to the staging directory:

   ```
   cd /home/user1
   cp 12100_linux_db.zip /stage/oracle/12.1.0
   ```

4. As the oracle user on the first node, unzip the Oracle media, as shown in the following example:

   ```
   cd /stage/oracle/12.1.0
   ```

```
unzip 12100_linux_db.zip
```

See Also:

- "Configuring Installation Directories and Shared Storage"
- "About Operating System Users and Groups"

Verifying My Oracle Support Credentials

When you first start OUI, you are prompted to enter your email address and My Oracle Support password. By entering this information, you enable the following features:

- Oracle Configuration Manager is installed and configured. This option enables you to associate information about your Oracle RAC configuration with your My Oracle Support account. In the event that you must place a service request with Oracle Support, that configuration information can help provide a more rapid resolution to the service issue.
- You can recieve e-mail notification of security alerts from My Oracle Support.
- Automatic download and application of the most recent patches to the newly installed Oracle software (with Oracle Grid Infrastructure for a cluster or Oracle Database). The software updates that can be downloaded include patches, critical patch updates, installer updates, and patch sets.

If you choose to enable these features, then you must supply your My Oracle Support account name (your email address) and your password. You may have to configure the proxy settings before your computer can connect to My Oracle Support.

If you have downloaded the software updates, then during installation you can enter the directory location where the files are stored on the local server instead of downloading them from My Oracle Support. The software updates are applied to the installed software during the installation process.

See Also:

- "About Credentials for Performing Backup and Recovery" on page 7-7
- "Obtaining the Patch" on page 10-3
- *Oracle Database 2 Day DBA*

Installing Oracle Grid Infrastructure for a Cluster

The software for Oracle Grid Infrastructure for a cluster consists of Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM).

The following topics describe the process of installing Oracle Grid Infrastructure for a cluster:

- Configuring the Operating System Environment
- About Cluster Verification Utility Fixup Scripts on Linux and UNIX
- Using Oracle Universal Installer to Install the Oracle Grid Infrastructure for a Cluster
- Completing the Oracle Clusterware Configuration

See Also:

- "Configuring Installation Directories and Shared Storage"
- "Verifying Operating System and Software Requirements"
- "About Hardware Requirements"
- "Creating the Oracle Grid Infrastructure for a Cluster Home Directory"

Configuring the Operating System Environment

You run OUI from the `oracle` user account. Before you start OUI to install Oracle Grid Infrastructure for a cluster, you do not have to configure the environment of the `oracle` user. You can specify the directories to use for the central inventory and the Grid home during the installation process.

However, you can set the `ORACLE_BASE` environment variable to the directory in which you want the Oracle Inventory files located. For example, if you plan to make the Oracle Database home directory `/u01/app/oracle/product/12.1.0/dbhome_1`, then you would set `ORACLE_BASE` to the directory `/u01/app/oracle/`. If you set the `ORACLE_BASE` directory before installation, then this becomes the default location for the central inventory displayed by OUI.

You can also set the `ORACLE_HOME` environment variable to the location chosen for the Grid home. If you set the `ORACLE_HOME` directory before installation, then this becomes the default location for the Grid home displayed by OUI.

(Optional) To modify the user environment before installing Oracle Grid Infrastructure for a cluster on Oracle Linux:

1. As the `oracle` user, execute the following commands:

```
[oracle]$ unset ORACLE_HOME
[oracle]$ unset ORACLE_SID
[oracle]$ unset ORACLE_BASE
[oracle]$ export ORACLE_BASE=/u01/app/oracle/
[oracle]$ export ORACLE_HOME=/u01/app/12.1.0/grid
```

2. Verify the changes have been made by executing the following commands:

```
[oracle]$ echo $ORACLE_SID

[oracle]$ echo $ORACLE_HOME
/u01/app/12.1.0/grid
[oracle]$ echo $ORACLE_BASE
/u01/app/oracle/
```

See Also:

- "About Operating System Users and Groups"
- "About Configuring the Software Owner's Shell Environment on Linux Systems"
- "Configuring Installation Directories and Shared Storage"

About Cluster Verification Utility Fixup Scripts on Linux and UNIX

During installation on Linux and UNIX platforms, for certain prerequisite check failures, you can click **Fix & Check Again** to generate a fixup script (`runfixup.sh`).

You are then prompted by OUI to run the fixup script as the root user in a separate session or you can specify through the OUI interface that the scripts should be run automatically. You must run the script on all the nodes specified by OUI.

Modifying the contents of the generated fixup script is not recommended.

Note: Using fixup scripts does not ensure that all the required prerequisites for installing Oracle Grid Infrastructure for a cluster and Oracle RAC are satisfied. You must still verify that all the requirements listed in Chapter 2, "Preparing Your Cluster" are met to ensure a successful installation.

See Also:

- "Installing Oracle Grid Infrastructure for a Cluster"
- "About Verifying the Oracle Clusterware Installation"

Using Oracle Universal Installer to Install the Oracle Grid Infrastructure for a Cluster

As the user that owns the software for Oracle Grid Infrastructure for a cluster (`oracle`)on the first node, install Oracle Grid Infrastructure for a cluster. Note that OUI uses Secure Shell (SSH) to copy the binary files from this node to the other nodes during the installation. OUI can configure SSH for you during installation.

Note: If you are installing Oracle Clusterware on a server that has a single-instance Oracle Database 11*g* installation, then stop the existing Oracle ASM instances, if any. After Oracle Clusterware is installed, start the Oracle ASM instances again. When you restart the single-instance Oracle database and the Oracle ASM instances, the Oracle ASM instances use the Cluster Synchronization Services Daemon (CSSD) instead of the daemon for the single-instance Oracle database.

To install the software for Oracle Grid Infrastructure for a cluster:

1. Use the following command to start OUI, where *staging_area* is the location of the staging area on disk, or the root level of the installation media:

   ```
   cd /staging_area/clusterware/Disk1
   ./runInstaller
   ```

   After a few minutes, the Download Software Updates window appears.

2. Choose to download the latest software updates, providing your registered My Oracle Support user name and password. You can use a staging location for the software updates or use software updates you have previously downloaded for this release. You can also proceed without checking for updates. Click **Next**.

   The Select Installation Option window appears.

3. Choose the **Install and Configure Grid Infrastructure for a Cluster** option, then click **Next**.

   The Select Cluster Type window appears.

4. Choose the **Configure a Standard cluster** option, then click **Next**.

   The Select Installation Type window appears.

5. Select **Typical Installation**, then click **Next**.

   The Specify Cluster Configuration window appears.

6. In the **SCAN Name** field, enter a name for your cluster that is unique throughout your entire enterprise network. For example, you might choose a name that is based on the node names' common prefix. This guide uses the SCAN name `docrac`.

   In the Hostname column of the table of cluster nodes, you should see your local node, for example racnode1.example.com. Click **Add** to add another node to the cluster.

   The Add Cluster Node Information pop-up window appears.

   > **Note:** Specify both nodes during installation even if you plan to use Oracle RAC One Node.

7. Enter the second node's public name (`racnode2`), and virtual IP name (`racnode2-vip`), and then click **OK**.

   You are returned to the Specify Cluster Configuration window.

8. You should now see both nodes listed in the table of cluster nodes. Click the **Identify Network Interfaces** button. In the Identify Network Interfaces window, verify that each interface has the correct interface type (Public or Private) associated with it. If you have network interfaces that should not be used by Oracle Clusterware, then set the network interface type to Do Not Use.

   Make sure both nodes are selected, then click the **SSH Connectivity** button at the bottom of the window.

   The bottom panel of the window displays the SSH Connectivity information.

9. Enter the operating system username and password for the Oracle software owner (`oracle`). Select the option **If you have configured SSH connectivity between the nodes**, then select the **Reuse private and public keys existing in user home** option. Click **Setup**.

   A message window appears, indicating that it might take several minutes to configure SSH connectivity between the nodes. After a short period, another message window appears indicating that passwordless SSH connectivity has been established between the cluster nodes. Click **OK** to continue.

10. When returned to the Specify Cluster Configuration window, click **Next** to continue.

    After several checks are performed, the Specify Install Locations window appears.

11. In the Specify Install Locations window, perform the following actions::

    - For the Oracle base field, make sure it is set to the location you chose for your Oracle base directory, for example `/u01/app/oracle/`. If not, then click **Browse**. In the Choose Directory window, go up the path until you can select the `/u01/app/oracle/` directory, then click **Choose Directory**.

    - For the Software Location field, make sure it is set to the location you chose for your Grid home, for example `/u01/app/12.1.0/grid`. If not, then click **Browse**. In the Choose Directory window, go up the path until you can select `/u01/app/12.1.0/grid`, then click **Choose Directory**.

- For the Cluster Registry Storage Type, choose **Automatic Storage Management**.
- Enter a password for a SYSASM user in the SYSASM Password and Confirm Password fields. This password is used for managing Oracle ASM after installation, so make note of it in a secure place.
- For the OSASM group, use the drop-down list to choose the operating system group for managing Oracle ASM, for example, `dba`.

After you have specified information for all the fields in this window, click **Next**.

The Create ASM Disk Group window appears.

12. In the **Disk Group Name** field, enter a name for the disk group, for example DATA. Choose the Redundancy level for this disk group, and then in the Add Disks section, choose the disks to add to this disk group.

    In the Add Disks section you should see the disks that you configured using the ASMLIB utility in "Using ASMLIB to Mark the Shared Disks as Candidate Disks" on page 2-27.

    When you have finished selecting the disks for this disk group, click **Next**.

    If you have not installed Oracle software previously on this computer, then the Create Inventory window appears.

13. Change the path for the inventory directory, if required. If you are using the same directory names as the examples in this book, then it should show a value of `/u01/app/oraInventory`. The group name for the `oraInventory` directory should show `oinstall`.

    > **Note:** The path displayed for the inventory directory should be the `oraInventory` subdirectory of the directory one level above the Oracle base directory. For example, if you set the `ORACLE_BASE` environment variable to `/u01/app/oracle/` before starting OUI, then the Oracle inventory path displayed is `/u01/app/oraInventory`.

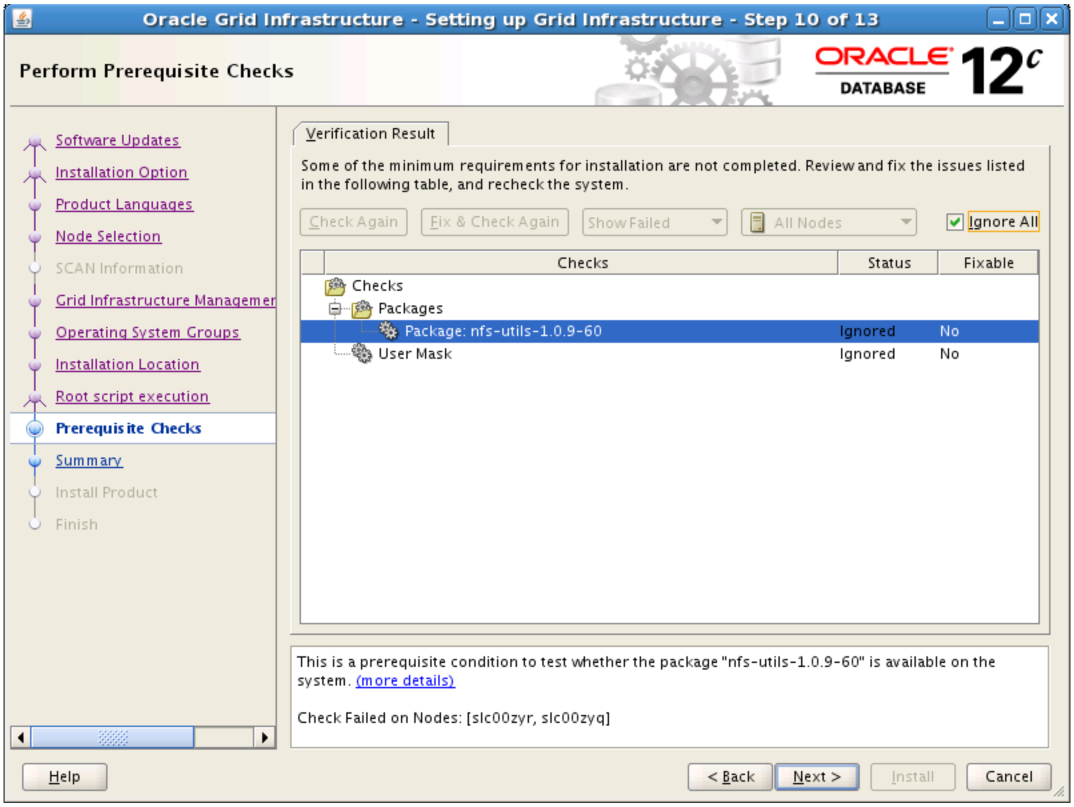
    Click **Next**. The Root Script Execution Configuration window appears.

14. 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 OUI.

    The Perform Prerequisite Checks window appears.

15. 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**.



Oracle Grid Infrastructure - Setting up Grid Infrastructure - Step 10 of 13

Perform Prerequisite Checks

Some of the minimum requirements for installation are not completed. Review and fix the issues listed in the following table, and recheck the system.

☑ Ignore All

Checks	Status	Fixable
Checks		
Packages		
Package: nfs-utils-1.0.9-60	Ignored	No
User Mask	Ignored	No

This is a prerequisite condition to test whether the package "nfs-utils-1.0.9-60" is available on the system. (more details)

Check Failed on Nodes: [slc00zyr, slc00zyq]

The Summary window appears.

16. Review the contents of the Summary window and then click **Finish**.

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

17. If you did not configure automation of the root scripts, then you are required to run certain scripts as the `root` user, as specified in the Execute Configuration Scripts window appears. Do *not* click **OK** until you have run the scripts. Run the scripts on all nodes as directed, in the order shown.

    For example, on Oracle Linux you perform the following steps (note that for clarity, the examples show the current user, node and directory in the prompt):

    a. As the `oracle` user on `racnode1`, open a terminal window, and enter the following commands:

    ```
    [oracle@racnode1 oracle]$ cd /u01/app/oraInventory
    [oracle@racnode1 oraInventory]$ su
    ```

    b. Enter the password for the `root` user, and then enter the following command to run the first script on `racnode1`:

    ```
    [root@racnode1 oraInventory]# ./orainstRoot.sh
    ```

    c. After the `orainstRoot.sh` script finishes on `racnode1`, open another terminal window, and as the `oracle` user, enter the following commands:

    ```
    [oracle@racnode1 oracle]$ ssh racnode2
    [oracle@racnode2 oracle]$ cd /u01/app/oraInventory
    [oracle@racnode2 oraInventory]$ su
    ```

    d. Enter the password for the `root` user, and then enter the following command to run the first script on `racnode2`:

```
[root@racnode2 oraInventory]# ./orainstRoot.sh
```

e. After the `orainstRoot.sh` script finishes on `racnode2`, go to the terminal window you opened in Step 15a. As the `root` user on `racnode1`, enter the following commands to run the second script, `root.sh`:

```
[root@racnode1 oraInventory]# cd /u01/app/12.1.0/grid
[root@racnode1 grid]# ./root.sh
```

Press Enter at the prompt to accept the default value.

Note: You must run the `root.sh` script on the first node and wait for it to finish. You can run `root.sh` scripts concurrently on all other nodes except for the last node on which you run the script. Like the first node, the `root.sh` script on the last node must be run separately.

---

f. After the `root.sh` script finishes on `racnode1`, go to the terminal window you opened in Step 15c. As the `root` user on `racnode2`, enter the following commands:

```
[root@racnode2 oraInventory]# cd /u01/app/12.1.0/grid
[root@racnode2 grid]# ./root.sh
```

After the `root.sh` script completes, return to the OUI window where the Installer prompted you to run the `orainstRoot.sh` and `root.sh` scripts. Click **OK**.

The software installation monitoring window reappears.

18. Continue monitoring the installation until the Finish window appears. Then click **Close** to complete the installation process and exit the installer.

Caution: After installation is complete, do not remove manually or run `cron` jobs that remove `/tmp/.oracle` or `/var/tmp/.oracle` directories or their files while Oracle software is running on the server. If you remove these files, then the Oracle software can encounter intermittent hangs. Oracle Clusterware installations can fail with the error:

```
CRS-0184: Cannot communicate with the CRS daemon.
```

See Also:

- "Installing Oracle Grid Infrastructure for a Cluster"
- "About Verifying the Oracle Clusterware Installation"

Completing the Oracle Clusterware Configuration

After you have installed Oracle Clusterware, verify that the node applications are started. Depending on which operating system you use, you may have to perform some postinstallation tasks to configure the Oracle Clusterware components properly.

To complete the Oracle Clusterware configuration on Oracle Linux:

1. As the `oracle` user on the first node, check the status of the Oracle Clusterware targets by entering the following command:

```
/u01/app/12.1.0/grid/bin/crsctl check cluster -all
```

This command provides output showing if all the required cluster services, such as `gsd`, `ons`, and `vip`, are started on the nodes of your cluster.

2. In the displayed output, you should see the Oracle Clusterware daemons are online for each node in the cluster.

```
**************************************************************
racnode1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
**************************************************************
racnode2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online

**************************************************************
```

If you see that one or more Oracle Clusterware resources are offline, or are missing, then the Oracle Clusterware software did not install properly.

Note: Avoid changing host names after you complete the Oracle Clusterware installation, including adding or deleting domain qualifications. Nodes with changed host names must be deleted from the cluster and added back with the new name.

See Also:

- "Installing Oracle Grid Infrastructure for a Cluster"
- "About Verifying the Oracle Clusterware Installation"

Installing the Oracle Database Software and Creating a Database

After Oracle Grid Infrastructure for a cluster is functional, you can install the Oracle Database software on the nodes of your cluster. OUI installs the software on the local node and then copies the binary files from the local node to all the other nodes in the cluster.

The following topics describe the process of installing the Oracle Database software and creating an Oracle RAC database or an Oracle RAC One Node database:

- Configuring the Operating System Environment
- Creating Additional Oracle ASM Disk Groups
- Using Oracle Universal Installer to Install Oracle RAC Software
- Verifying Your Oracle RAC Database Installation

See Also:

- "Installing Oracle Grid Infrastructure for a Cluster" on page 3-3
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC" on page 1-3
- "Configuring Installation Directories and Shared Storage" on page 2-21

Configuring the Operating System Environment

You run OUI from the `oracle` user account. Before you start OUI you do not have to configure the environment of the `oracle` user.

However, you can set the `ORACLE_BASE` environment variable to the directory in which you want the Oracle Inventory files located. For example, if you plan to make the Oracle Database home directory `/u01/app/oracle/product/12.1.0/dbhome_1`, then you would set `ORACLE_BASE` to the directory `/u01/app/oracle/`. If you set the `ORACLE_BASE` directory before installation, then this becomes the default location for the central inventory displayed by OUI.

You can also set the `ORACLE_HOME` environment variable to the location chosen for the Oracle Database home. If you set the `ORACLE_HOME` directory before installation, then this becomes the default location for the Grid home displayed by OUI.

(Optional) To modify the user environment before installing Oracle Database software on Oracle Linux:

1. As the `oracle` user, execute the following commands:

```
[oracle]$ unset ORACLE_SID
[oracle]$ export ORACLE_BASE=/u01/app/oracle/
[oracle]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
```

2. Verify the changes have been made by executing the following commands:

```
[oracle]$ echo $ORACLE_SID

[oracle]$ echo $ORACLE_HOME
/u01/app/oracle/product/12.1.0/dbhome_1
[oracle]$ echo $ORACLE_BASE
/u01/app/oracle/
```

3. Install Oracle Grid Infrastructure for a cluster. See “Installing Oracle Grid Infrastructure for a Cluster” on page 3.

See Also:

- "About Operating System Users and Groups"
- "About Configuring the Software Owner's Shell Environment on Linux Systems"
- "Configuring Installation Directories and Shared Storage"

Creating Additional Oracle ASM Disk Groups

If you chose to store the Oracle Clusterware files on Oracle ASM during the Oracle Grid Infrastructure for a cluster installation, then a single disk group was created in Oracle ASM. You can use this same disk group to store the data files for your Oracle database.

If you want to use a separate disk group for your Oracle database files or for the fast recovery area, then you must create the additional Oracle ASM disk groups before installing Oracle Database software.

To create an additional disk group using ASMCA:

1. Prepare the disks or devices for use with Oracle ASM, as described in "Configuring Installation Directories and Shared Storage" on page 2-21.
2. Start the Oracle Automatic Storage Configuration Assistant (ASMCA) from the Grid home:

   ```
   /u01/app/12.1.0/grid/bin/asmca
   ```

   ASMCA starts, and displays the Disk Groups window.
3. Click the **Create** button to create a disk group.

   The Create Disk Group window appears.
4. Provide the following information:

   - In the **Disk Group Name** field, enter a name for the new disk group, for example, FRA.
   - Choose a **Redundancy** level, for example, Normal.
   - Select the disks to include in the new disk group.

     If you used ASMLIB to configure the disks for use with Oracle ASM, then the available disks are displayed if you have the Show Eligible option selected, and they have a Header Status of PROVISIONED.

   After you have provided all the information, click **OK**. A progress window titled DiskGroup: Creation appears. After a few minutes, a message appears indicating the disk group was created successfully. Click **OK** to continue.
5. Repeat Step 3 and 4 to create additional disk groups, or click **Exit**, then select **Yes** to exit the utility.

   See Also:

   - "About Oracle Automatic Storage Management" on page 1-4
   - "Configuring Files on an NAS Device for Use with Oracle ASM" on page 2-26
   - "Using ASMLIB to Mark the Shared Disks as Candidate Disks" on page 2-27

Using Oracle Universal Installer to Install Oracle RAC Software

After you have configured the operating system environment, you can use Oracle Universal Installer to install the Oracle RAC software and create an Oracle RAC database. If you plan to use Oracle RAC One Node, then you must first install the Oracle RAC software without creating a database. After the installation completes, you use Database Configuration Assistant (DBCA) to create the Oracle RAC One Node database.

To install Oracle Database software on your cluster and to create a database:

1. As the `oracle` user, use the following commands to start OUI, where *`staging_area`* is the location of the staging area on disk, or the location of the mounted installation disk:

```
cd /staging_area
./runInstaller
```

The Configure Security Updates window appears.

2. (Optional) Enter your email address and My Oracle Support password, then click **Next** to continue.

   If you want to receive notification by email of any newly discovered security issues related to the software you are installing, then enter an email address in the **Email** field. If you also want to receive the security updates through My Oracle Support, then use the same email address that is registered with My Oracle Support, select the **I wish to receive security updates via My Oracle Support** option, and enter your My Oracle Support login password in the **My Oracle Support Password** field.

   If you provide an email address, then the Oracle Configuration Manager (OCM) tool will also be installed. This utility provides Oracle Support Services with configuration information about your system when creating service requests. You can disable the OCM tool after installation, but Oracle strongly discourages this. OCM does not access, collect, or store any personal information (except for technical support contact information), or any business data files residing in your software environment. For more information about OCM, see `http://www.oracle.com/technetwork/documentation/ocm-092152.html`.

   After you click **Next**, the Select Installation Option window appears.

3. If you want to create an Oracle RAC database, then select **Create and configure a database**. If you want to create an Oracle RAC One Node database, then select **Install database software only**. Click **Next** to continue.

   The System Class window appears.

4. Choose **Server Class**, then click **Next**.

   If you choose the Desktop Class option, then OUI installs a single-instance database, not a clustered database.

   The Node Selection window appears.

5. Select the **Real Application Clusters database** installation type.

   Select the nodes on which you want to install Oracle Database software and create an Oracle RAC instance. All the available nodes in the cluster are selected by default.

   Note: Select both nodes during installation, even if you are creating an Oracle RAC One Node database.

   Click the **SSH Connectivity** button at the bottom of the window. The bottom panel of the window displays the SSH Connectivity information.

6. Because you configured SSH connectivity between the nodes for the Oracle Grid Infrastructure for a cluster installation, select the **Reuse private and public keys existing in user home option**. If you are using a network user that has a home directory on shared storage, then also select the **User home if shared by the selected nodes** option. Click **Test**.

   A message window appears, indicating that passwordless SSH connectivity has been established between the cluster nodes. Click **OK** to continue.

7. When returned to the Node Selection window, click **Next** to continue.

   The Select Install Type window appears.

8. Choose the **Typical install** option, and click **Next**.

   A typical installation requires minimal input. It installs the software and optionally creates a general-purpose database. If you choose the Advanced installation type (not documented in this guide), then you are prompted to provide more information about how the database should be configured. For example, you could set passwords for each user account individually, choose a different template for the starter database, choose a nondefault language for the database, and so on.

   The Typical Install Configuration window appears.

9. In this window, you must provide the following information:

   - **Oracle base location**: The default value is `/u01/app/oracle/`. If you did not set the `ORACLE_BASE` environment variable and the default location is different from the directory location you have chosen, then enter the directory for your Oracle base or click the Browse button to change the directory path.

   - **Software location**: If you did not set the `ORACLE_HOME` environment variable before starting the installation, then enter the directory for your Oracle home or click the Browse button to change the directory path.

   - **Storage Type**: In this drop-down list, choose Automatic Storage Management (ASM). If you do not want to use Oracle ASM, then choose File System. Because Oracle ASM was installed with the Oracle Grid Infrastructure for a cluster, Oracle Automatic Storage Management is the default value.

   - **Database file location**: Choose the disk group to use for storing the database files. You can use the same disk group that Oracle Clusterware uses. If you do not want to use the same disk group that is currently being used to store the Oracle Clusterware files, then you must exit the installation and create a new disk group using Oracle ASM utilities. Refer to "Creating Additional Oracle ASM Disk Groups" on page 3-11 for more information on creating a disk group.

     If you chose the File System storage type, then enter the directory location of the shared storage where the database files will be created.

   - **ASMSNMP Password**: Enter the password for the ASMSNMP user. The ASMSNMP user is used primarily by Oracle Enterprise Manager to monitor Oracle ASM instances. See *Oracle Automatic Storage Management Administrator's Guide* for more information about the ASMSNMP user.

     If you chose File System for the Storage Type, then this field is disabled.

   - **Database edition**: From this drop-down list choose either Enterprise Edition or Standard Edition. The number in parentheses next to your choice indicates the amount of disk space required.

   - **OSDBA Group**: From this drop-down list select the operating system group used for database administration, for example, `dba`.

   - **Global database name**: Enter the fully qualified global name for your database. The global database name is in the form *`DB_UNIQUE_NAME.DB_DOMAIN`*, for example, `orcl.example.com`.

> **Note:** The first eight characters of the `ORACLE_SID` must be a unique identifier for each database. The SID prefix cannot contain underscore (_), dollar ($), or pound (#) characters. The total length of the `ORACLE_SID` cannot exceed 15 characters for Oracle RAC databases.

- **Administrative password**: Enter the password to be used for the administrative account, such as SYS, SYSTEM, and DBSNMP.
- **Confirm Password**: Enter the same password in this field.

After you have provided all the necessary information, click **Next**. The Perform Prerequisite Checks window appears.

10. After a short period, the Summary window appears. Review the information on this window, then click **Finish** to continue.

    If any of the information in the Summary window is incorrect, then use the Back button to return to a previous window and correct it.

    After you click Finish, OUI displays a progress indicator to show that the installation has begun. This step takes several minutes to complete.

11. After the software is installed on each node, if you select the option to create a database, then OUI starts the Database Configuration Assistant (DBCA). This utility creates the database using the global database name specified in Step 9. At the end of the database creation, you see the DBCA window with the database configuration information displayed.

    There is also a Password Management button that you can click to unlock the database user accounts, or change their default passwords.

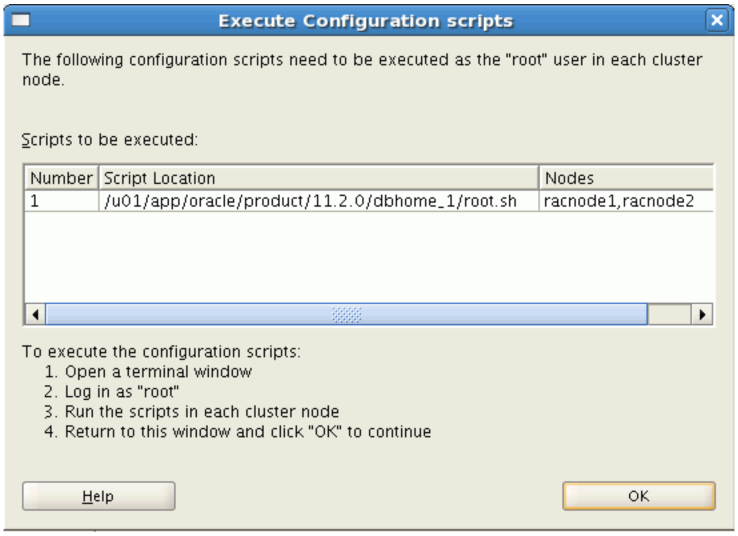
    After making note of the information in the Summary window, click **OK**.

    OUI configures Oracle Configuration Management, if you provided the information in Step 2.

    If you chose to perform a software-only installation, then the database configuration assistants are not started. You must run DBCA separately to create the Oracle RAC One Node database.

    > **See Also:** *Oracle Real Application Clusters Installation Guide* for your platform for information about creating an Oracle RAC One Node database using DBCA

12. In the last step of the installation process, you are prompted to perform the task of running the `root.sh` script on both nodes, as specified in the Execute Configuration Scripts window. Do *not* click **OK** until you have run the scripts on all nodes.



The following configuration scripts need to be executed as the "root" user in each cluster node.

Scripts to be executed:

Number	Script Location	Nodes
1	/u01/app/oracle/product/11.2.0/dbhome_1/root.sh	racnode1,racnode2

To execute the configuration scripts:
1. Open a terminal window
2. Log in as "root"
3. Run the scripts in each cluster node
4. Return to this window and click "OK" to continue

Perform the following steps to run the root.sh script (note that for clarity, the examples show the current user, node and directory in the prompt):

a. Open a terminal window as the oracle user on the first node. Change directories to your Oracle home directory, and then switch to the root user by entering the following commands:

```
[oracle@racnode1 oracle]$ cd /u01/app/oracle/product/12.1.0/dbhome_1
[oracle@racnode1 dbhome_1]$ su
```

b. Enter the password for the root user, and then run the script specified in the Execute Configuration scripts window:

```
[root@racnode1 dbhome_1]# ./root.sh
```

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

---

c. As the root.sh script runs, it prompts you for the path to the local bin directory. The information displayed in the brackets is the information it has obtained from your system configuration. It also writes the dbhome, oraenv, and coraenv files in the /usr/local/bin directory. If these files exist, then you are asked to overwrite them. After responding to prompt, press the **Enter** key. To accept the default choice, press the Enter key without entering any text.

d. Enter commands similar to the following to run the script on the other nodes:

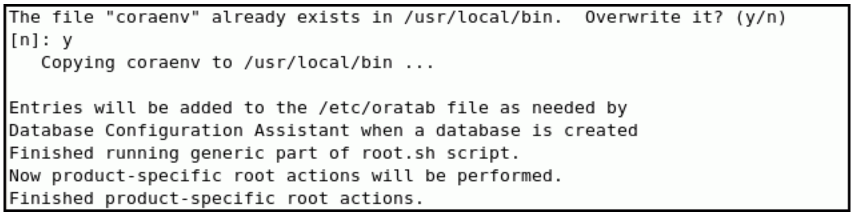
```
[root@racnode1 dhome_1]# exit
[oracle@racnode1 dhome_1]$ ssh racnode2
[oracle@racnode2 ~]$ cd /u01/app/oracle/product/12.1.0/dbhome_1
[oracle@racnode2 dbhome_1]$ su
```

e. Enter the password for the root user, and then run the script specified in the Execute Configuration scripts window:

```
[root@racnode2 dbhome_1]# ./root.sh
```

f. After responding to each prompt, press the **Enter** key.

g. When the root.sh script completes, the following messages are displayed:



```
The file "coraenv" already exists in /usr/local/bin.  Overwrite it? (y/n)
[n]: y
   Copying coraenv to /usr/local/bin ...

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
Finished product-specific root actions.
```

After you finish executing the script on all nodes, return to the Execute Configuration scripts window and click **OK**.

The Install Product window displays.

13. Click **Next** to complete the installation.

    The Finish window is displayed.

14. Click **Close** to exit the installer.

See Also:

- "Configuring the Operating System Environment"
- "Verifying Your Oracle RAC Database Installation"
- "Performing Postinstallation Tasks"
- "About Downloading and Installing Oracle Database Patches"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring disk groups in Oracle ASM

Verifying Your Oracle RAC Database Installation

If you chose to create an Oracle RAC database during installation, then verify that all the database services are up and running.

To verify the Oracle RAC database services are started:

1. Log in as the `oracle` user and go to the *`Grid_home`*`/bin` directory:

```
[oracle] $ cd /u01/app/12.1.0/grid/bin
```

2. Run the following command to view the status of the resources managed by Oracle Clusterware that contain the string 'ora':

```
[oracle] $ ./crsctl status resource -w "TYPE co 'ora'" -t
```

The output of the command should show that the Oracle Clusterware, Oracle ASM, and Oracle Database resources are available (online) for each host. An example of the output is:

```
------------------------------------------------------------------------------
NAME                TARGET STATE    SERVER             STATE_DETAILS
------------------------------------------------------------------------------
Local Resources
------------------------------------------------------------------------------
ora.DATA.dg
                    ONLINE  ONLINE   racnode1
                    ONLINE  ONLINE   racnode2
```

```
ora.LISTENER.lsnr
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2

ora.asm

                  ONLINE  ONLINE   racnode1          Started
                  ONLINE  ONLINE   racnode2          Started

ora.eons
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2

ora.gsd
                  OFFLINE OFFLINE   racnode1
                  OFFLINE OFFLINE   racnode2

ora.net1.network
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2

ora.ons
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2
ora.registry.acfs
                  ONLINE  ONLINE   racnode1
                  ONLINE  ONLINE   racnode2
------------------------------------------------------------------------------
Cluster Resources
------------------------------------------------------------------------------
ora.LISTENER_SCAN1.lsnr
      1           ONLINE  ONLINE   racnode1
ora.oc4j
      1           OFFLINE OFFLINE
ora.orcl.db
      1           ONLINE  ONLINE   racnode1            Open
      2           ONLINE  ONLINE   racnode2            Open
ora.racnode1.vip
      1           ONLINE  ONLINE   racnode1
ora.racnode2.vip
      1           ONLINE  ONLINE   racnode2
ora.scan1.vip
      1           ONLINE  ONLINE   racnode1
```

Caution: After installation is complete, do not remove manually or run `cron` jobs that remove `/tmp/.oracle` or `/var/tmp/.oracle` directories or their files while Oracle software is running on the server. If you remove these files, then the Oracle software can encounter intermittent hangs. Oracle Clusterware installations will fail with the error:

CRS-0184: Cannot communicate with the CRS daemon.

See Also:

- "Installing the Oracle Database Software and Creating a Database"
- "Performing Postinstallation Tasks"
- "About Downloading and Installing Oracle Database Patches"
- "About Converting an Oracle Database to an Oracle RAC Database"

Performing Postinstallation Tasks

After you have installed the Oracle Real Application Clusters (Oracle RAC) software and created an Oracle RAC database there are additional tasks to perform before your cluster database is ready for use. These steps are recommended, but are not required.

This section contains the following topics:

- About Verifying the Oracle Clusterware Installation
- Creating an Oracle RAC Database Using DBCA
- Backing Up Installation Files
- About Downloading and Installing Oracle Database Patches
- About Configuring User Accounts

See Also:

- "Completing the Oracle Clusterware Configuration"
- "Verifying Your Oracle RAC Database Installation"

About Verifying the Oracle Clusterware Installation

After the Oracle Clusterware installation is complete, OUI automatically runs Cluster Verification Utility (CVU) as a configuration assistant to verify that the Oracle Clusterware installation has been completed successfully.

If CVU reports problems with your configuration, then correct these errors before proceeding.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for more information about using CVU and resolving configuration problems
- "Verifying Your Oracle RAC Database Installation" on page 3-17

Creating an Oracle RAC Database Using DBCA

If you did not select the option to create an Oracle RAC database during installation, then you will need to create one using DBCA after you have verified that the installation of the Oracle RAC software was successful. The steps for creating an Oracle RAC database are documented in *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

Backing Up Installation Files

Certain files used during installation are very important to the operation of the installed software. It is important to back up these files and keep them in a separate location from the installed software in case of hardware failure.

- About Backing Up the root.sh Script

See Also:

- "Overview of Oracle RAC Database Backup and Recovery" on page 7-1
- "Performing Backups of Your Oracle Real Application Clusters Database" on page 7-7
- "Performing Postinstallation Tasks"

About Backing Up the root.sh Script

Oracle recommends that you back up the `root.sh` script after you complete an installation. If you install other products in the same Oracle home directory, then OUI updates the contents of the existing `root.sh` script during the installation. If you require information contained in the original `root.sh` script, then you can recover it from the `root.sh` backup copy.

See Also:

- "Using Oracle Universal Installer to Install the Oracle Grid Infrastructure for a Cluster" on page 3-5
- "Performing Postinstallation Tasks"

About Downloading and Installing Oracle Database Patches

Periodically, Oracle issues bug fixes for its software called **patches**. **Patch sets** are a collection of bug fixes that were produced up to the time of the patch set release. Patch sets are fully tested product fixes. Application of a patch set affects the software residing in your Oracle home.

Ensure that you run the latest patch set of the installed software. If you configured access to My Oracle Support during installation, then the latest patches should have been downloaded and applied during installation.

If you did not configure access to My Oracle Support within OUI, then you should apply the latest patch set for your release and any necessary patches that are not included in a patch set. Information about downloading and installing patches and patch sets is covered in Chapter 10, "Managing Oracle Software and Applying Patches".

See Also:

- "Verifying My Oracle Support Credentials" on page 3-3
- "Obtaining the Patch" on page 10-3
- *Oracle Grid Infrastructure Installation Guide for Linux* for detailed instructions on how to locate and download patches and patch sets
- *Oracle Universal Installer NextGen Installation Guide*

About Configuring User Accounts

The oracle user operating system account is the account that you used to install the Oracle software. You can use different operating system accounts for accessing and managing your Oracle RAC database. You can modify the shell configuration file to set environment variables such as ORACLE_HOME whenever you log in as that operating system user.

See Also:

- "About Operating System Users and Groups" on page 2-9
- "Installing the Oracle Database Software and Creating a Database" on page 3-10
- "Performing Postinstallation Tasks" on page 3-19
- *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems* for more information about setting up optional operating system user accounts you use to manage the database

About Converting an Oracle Database to an Oracle RAC Database

You can use rconfig or Oracle Enterprise Manager to assist you with the task of converting a single-instance database installation to an Oracle Real Application Clusters (Oracle RAC) database. rconfig is a command line utility. The Convert to Cluster Database option in Oracle Enterprise Manager Cloud Control provides a GUI conversion tool. Additionally, after you have converted your single-instance database to an Oracle RAC database, you can use the srvctl utility to convert the database to an Oracle RAC One Node database.

This section contains the following topics:

- About Preparing for Database Conversion
- Overview of the Database Conversion Process Using Cloud Control
- Overview of the Database Conversion Process Using rconfig
- Converting an Oracle RAC Database into an Oracle RAC One Node Database

See Also:

- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC" on page 1-3
- "Verifying System Requirements" on page 2-1
- "Preparing the Server" on page 2-9
- "Configuring Installation Directories and Shared Storage" on page 2-21

About Preparing for Database Conversion

Before you start the process of converting your database to a cluster database, your database environment must meet certain prerequisites.

- The existing database and the target Oracle RAC database must be on the same release of Oracle Database 12*c* and must be running on the same platform.
- The hardware and operating system software used to implement your Oracle RAC database must be certified for use with the release of the Oracle RAC software you are installing.

- You must configure shared storage for your Oracle RAC database.
- You must verify that any applications that run against the Oracle RAC database do not need any additional configuration before they can be used successfully with the cluster database. This applies to both Oracle applications and database features, such as Oracle Streams, and applications and products that do not come from Oracle.
- Backup procedures should be available before converting from a single-instance Oracle Database to Oracle RAC.
- For archiving in Oracle RAC environments, the archive log file format requires a thread number.
- The archived redo log files from all instances of an Oracle RAC database are required for media recovery. If you archive to a file and you do not use a cluster file system, or some other means to provide shared file systems, then you require a method of accessing the archived redo log files from all nodes on which the cluster database has instances.

Note: For information about using individual Oracle Database 12*c* database products or options, refer to the product documentation library, which is available on the OTN website at `http://www.oracle.com/technetwork/indexes/documentation/`

See Also:

- "About Installing Oracle RAC on Different Operating Systems" on page 1-8
- "Verifying System Requirements"
- "Configuring Installation Directories and Shared Storage"
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"

Overview of the Database Conversion Process Using Cloud Control

This section summarizes the process of converting a single-instance database to an Oracle RAC database using Oracle Enterprise Manager Cloud Control:

- Complete the prerequisite tasks for converting to an Oracle RAC database:
  - Oracle Clusterware and Oracle Database software is installed on all target nodes.
  - Oracle Clusterware is started.
  - The Oracle Database binary is enabled for Oracle RAC on all target nodes.
  - Shared storage is configured and accessible from all nodes.
  - User equivalency is configured for the operating system user performing the conversion.
  - Enterprise Manager agents are configured and running on all nodes, and are configured with the cluster and host information.
  - The database being converted has been backed up successfully.

- Access the Database Home page for the database you want to convert. For instructions on navigating to the Database Home page, see "Accessing the Database Home Page" in the *2 Day DBA Guide*.
- From the **Availability** menu, select **Convert to Cluster Database**.
- Provide the necessary credentials.
- Select the host nodes that should contain instances of the new database.
- Provide listener and instance configuration information.
- Specify the location of the shared storage to be used for the data files.
- Submit the job.
- Complete the post-conversion tasks.

The resulting Oracle RAC database uses a server pool instead of a fixed configuration.

> **See Also:** *Oracle Real Application Clusters Installation Guide* for your platform for a complete description of this process

Overview of the Database Conversion Process Using rconfig

The following list provides an outline of the process of converting a single-instance database to an Oracle RAC database using the `rconfig` utility:

- Complete the prerequisite tasks for converting to an Oracle RAC database.
  - Oracle Clusterware and Oracle Database software is installed on all target nodes.
  - Oracle Clusterware is started.
  - The Oracle Database binary is enabled for Oracle RAC on all target nodes.
  - Shared storage is configured and accessible from all nodes.
  - User equivalency is configured for the operating system user performing the conversion.
  - The database being converted has been backed up successfully.
- Using `rconfig`, you can convert a single-instance database to either an administrator-managed cluster database or a policy-managed cluster database. Modify the parameters in either the `ConvertToRAC_AdminManaged.xml` or `ConvertToRAC_PolicyManaged.xml` sample file, as appropriate for your environment, then save the file to a new location. Both files are located in the *`Oracle_home`*`/assistants/rconfig/sampleXMLs` directory.
- Run the `rconfig` command, supplying the name of the modified XML file as input.
- Complete the post-conversion tasks.

You can also use the `rconfig` utility to convert single-instance Oracle ASM to clustered Oracle ASM.

> **See Also:** *Oracle Real Application Clusters Installation Guide* for your platform for a complete description of this process

Converting an Oracle RAC Database into an Oracle RAC One Node Database

After you use the `rconfig` utility to convert a single-instance Oracle database into a single-node Oracle RAC database, you can use the `srvctl` utility to convert the database into an Oracle RAC One Node database.

To convert your database to an Oracle RAC One Node database, use the following command:

```
srvctl convert database -db <database_name> -dbtype RACONENODE
```

An Oracle RAC One Node database must be part of a multi-node cluster to support failover or online database relocation. You must either install Oracle Grid Infrastructure for a cluster and Oracle RAC on at least two nodes, or add a node to your existing single-node Oracle RAC database.

See Also:

- Chapter 9, "Adding and Deleting Nodes and Instances" for more information about adding nodes
- "About Oracle RAC One Node" for more information about Oracle RAC One Node

4 Administering Database Instances and Cluster Databases

Web-based Oracle Enterprise Manager Cloud Control enables you to manage Oracle Real Application Clusters (Oracle RAC) databases. The Enterprise Manager console is a central point of control for the Oracle environment. You can use the Cloud Control console to administer multiple Oracle RAC databases and cluster nodes.

The following topics describe how to administer your Oracle RAC environment, including starting and stopping database components and how to administer parameters and parameter files in Oracle RAC:

- About Oracle Real Application Clusters Database Management
- About Oracle RAC One Node Database Management
- About Oracle RAC Management Using Enterprise Manager
- Starting and Stopping Oracle RAC Databases and Database Instances
- About Oracle Real Application Clusters Initialization Parameters
- About Administering Storage in Oracle RAC

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*
- "Monitoring Oracle RAC Database and Cluster Performance"
- "About the Fast Recovery Area in Oracle RAC"
- "Performing Backups of Your Oracle Real Application Clusters Database"

About Oracle Real Application Clusters Database Management

Oracle Real Application Clusters (Oracle RAC) provides technology that links two or more individual computers so that they function as one system. Oracle RAC enables each **node**, which is a computer that is a member of a cluster, to share access to the Oracle database. If one cluster node fails or is taken offline, then the other cluster nodes continue operating and the entire Oracle RAC database remains available. Two or more inexpensive computers can appear to applications as if they were a single, much more powerful and more expensive, computer.

To increase the performance of an Oracle RAC database, you can add cluster nodes. Each additional node can help speed up application processing, support more users or processes, or both. In addition, you can also add cluster nodes to increase the availability and reliability of a two-node Oracle RAC database. The more nodes that your Oracle RAC environment has, the less the impact of the loss of any individual node on the database.

> **Note:** If you are using Oracle Database Standard Edition, then your cluster must adhere to the license restrictions. See *Oracle Database Licensing Information* for specific details on these restrictions.

An Oracle RAC database requires three components: cluster nodes, shared storage, and Oracle Clusterware. Although you can choose how many nodes your cluster should have and what type of shared storage to use, this guide describes one specific configuration for a two-node cluster. This two-node configuration uses Oracle Automatic Storage Management (Oracle ASM) for storage management and Recovery Manager (RMAN) for the backup and recovery strategy.

Most administration tasks are the same for Oracle single-instance and Oracle RAC databases. This guide provides additional instructions for database administration tasks specific to Oracle RAC, and recommendations for managing Oracle RAC databases.

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA* for information about managing single instance databases
- Chapter 9, "Adding and Deleting Nodes and Instances"

About Oracle RAC One Node Database Management

Oracle RAC One Node databases are administered slightly differently from Oracle RAC or single-instance databases. For administrator-managed Oracle RAC One Node databases, you must monitor the candidate node list and make sure a server is always available for failover if possible. Candidate servers reside in the Generic server pool of the cluster. If the node currently hosting the Oracle RAC One node database instance fails, then the database and its services fail over to one of the candidate servers. For policy-managed Oracle RAC One Node databases, you must ensure that the server pools are configured such that a server will be available for the database to fail over to in case its current node becomes unavailable. Also, for policy-managed Oracle RAC One Node databases, the destination node for online database relocation must be located in the database's server pool.

Oracle Real Application Clusters One Node (Oracle RAC One Node) is a single instance of an Oracle Real Application Clusters (Oracle RAC) database that runs on one node in a cluster. Instead of stopping and starting instances, you use Oracle RAC One Node online database relocation to relocate the Oracle RAC One Node instance to another server.

See Also: *Oracle Real Application Clusters Administration and Deployment Guide* for more information about managing Oracle RAC One Node databases

About Oracle RAC Management Using Enterprise Manager

The web-based Oracle Enterprise Manager Cloud Control console enables you to manage Oracle RAC and Oracle RAC One Node databases. Enterprise Manager is a central point of control for the Oracle environment that you access by way of a graphical user interface (GUI). You can use Enterprise Manager to create and modify services, and to start and stop the cluster database instances and the cluster database. Use Enterprise Manager Cloud Control to administer your entire Oracle RAC environment, not just the Oracle RAC database.

When you log in to Enterprise Manager and access the database home page for an Oracle RAC database, the Cluster Database Home page appears. The Cluster Database Home page is similar to a single-instance Database Home page. However, on the Cluster Database Home page, Enterprise Manager displays the system state and availability of the entire Oracle RAC environment. This includes a summary about alert messages and job activity, and the status of and links to all the database and Oracle Automatic Storage Management (Oracle ASM) instances. By clicking the Cluster tab on this page you can view the Cluster Home page to view the status of and alerts for the underlying cluster.

Note: Oracle Enterprise Manager Cloud Control manages all the Oracle ASM instances on the nodes. Only one Oracle ASM instance exists per node, no matter how many Oracle databases use Oracle ASM for storage.

See Also:

- *Oracle Database 2 Day DBA* for information about using Oracle EM Database Express and accessing the Database Home page
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about monitoring Oracle RAC performance

Starting and Stopping Oracle RAC Databases and Database Instances

Typically, you start and shut down the cluster database from the Enterprise Manager Cluster Database Home page. By using this page for cluster database startup and shutdown operations, you ensure that all the instances that belong to the Oracle RAC database are in a consistent state. Oracle Enterprise Manager enables you to more easily manage an Oracle RAC database.

You can start and stop individual instances in an Oracle RAC database. Starting and stopping one instance in an Oracle RAC database does not stop or start the other database instances. To completely stop an Oracle RAC database, you must shut down all of its instances.

To start and stop an entire Oracle RAC database, assuming you are using a server parameter file (SPFILE):

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, logging in as user `SYS`.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. In the Cluster Database Home page, in the General section, click **Startup** if the database is down, or **Shutdown** if the database is started.

   The Startup/Shutdown: Specify Credentials page appears.

3. Enter the host credentials for the cluster nodes. The host credentials are the user name and password for a user who is a member of the `OSDBA` or `OSOPER` operating system group.

   The Startup/Shutdown: Select Operation page appears.

4. Click **Select All** to select all the instances, or then click **Shutdown** to stop all the database instances or **Startup** to start all the database instances.

   The Startup/Shutdown: Confirmation page appears.

5. Click **Yes**.

To start and stop individual instances, go to the Startup/Shutdown: Select Operation page and select the database instances, then click Startup or Shutdown to perform the desired operation on the selected database instances. You can also start and shut down instances using SQL*Plus or Server Control (SRVCTL).

> **Note:** You can start and shut down individual instances from each instance's home page. However, it is easier to perform instance startup and shutdown operations directly from the Startup/Shutdown: Select Operation page.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about using command-line interfaces to start and stop Oracle RAC database instances

About Oracle Real Application Clusters Initialization Parameters

Managing initialization parameters for an Oracle RAC database is essentially the same as managing them for a single-instance Oracle database. Note the following differences for parameters in Oracle RAC databases:

- Parameters that are cluster-specific have the value Cluster Database in the Category column.
- Parameters that are the same on each instance in the Oracle RAC database are identified in the Instance column with an asterisk (*).
- Parameters that are set to different values on each instance of an Oracle RAC database are listed by instance name.

One difference when managing initialization parameters in an Oracle RAC environment is that parameters can apply to more than one instance of the database. If you change a parameter setting that is marked by an asterisk, which indicates that the parameter is a clusterwide database initialization parameter, then you change that parameter's setting for all the instances in your Oracle RAC database. If you change an initialization parameter prefixed with an instance name, or an instance-specific initialization parameter, then the change affects only that instance; the change does not affect the parameter's settings on other database instances.

This section contains the following topics:

- About Configuring Initialization Parameters for an Oracle RAC Database

- Editing Initialization Parameter Settings for an Oracle RAC Database
- About the Server Parameter File for Oracle Real Application Clusters

See Also:

- *Oracle Database 2 Day DBA* for more information about viewing and modifying the initialization parameter settings for your database
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in an Oracle RAC environment

About Configuring Initialization Parameters for an Oracle RAC Database

A server parameter file (SPFILE) is a type of repository for initialization parameters that is maintained on the server where the Oracle database server runs, or on shared storage for an Oracle RAC database. Initialization parameters stored in a server parameter file are persistent, in that any changes made to the parameters while an instance is running can persist across instance shutdown and startup.

An initialization parameter file (PFILE) is a text file that contains initialization parameter settings. In contrast to the SPFILE, this parameter file is not binary and does not need to be located on the database server. The text-based initialization parameter file can be read by the database, but it is not written to by the database.

By default, Oracle Database sets most parameters to a default value and this value is the same across all instances. However, many initialization parameters can also have different values on different instances as described in *Oracle Database Reference.* Other parameters *must* be either unique or identical across instances, as described in the following sections:

- Parameters that Must Have Identical Settings on All Instances
- Parameters that Must Have Unique Settings on All Instances
- Parameters that Should Have Identical Settings on All Instances

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in Oracle RAC environments

Parameters that Must Have Identical Settings on All Instances

Certain initialization parameters that are critical at database creation or that affect certain database operations must have the same value for every instance in an Oracle RAC database. Specify these parameter values in the SPFILE, or within the individual PFILEs for each instance. The following list contains the parameters that must be identical on every instance:

- `CLUSTER_DATABASE`
- `COMPATIBLE`
- `CONTROL_FILES`
- `DB_BLOCK_SIZE`
- `DB_DOMAIN`

- DB_FILES
- DB_NAME
- DB_RECOVERY_FILE_DEST
- DB_RECOVERY_FILE_DEST_SIZE
- DB_UNIQUE_NAME
- DML_LOCKS (only if set to 0)
- INSTANCE_TYPE (RDBMS or ASM)
- PARALLEL_EXECUTION_MESSAGE_SIZE
- REMOTE_LOGIN_PASSWORDFILE
- RESULT_CACHE_MAX_SIZE (either enabled or disabled on all instances)
- UNDO_MANAGEMENT

> **See Also:**
>
> - *Oracle Database 2 Day DBA*
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in Oracle RAC environments

Parameters that Must Have Unique Settings on All Instances

For the following parameters, the parameter must be set on all instances to a value that is unique to that instance:

- ASM_PREFERRED_READ_FAILURE_GROUPS
- CLUSTER_INTERCONNECTS
- INSTANCE_NUMBER
- ROLLBACK_SEGMENTS (if the UNDO_MANAGEMENT initialization parameter is not set to AUTO)
- UNDO_TABLESPACE (if UNDO_MANAGEMENT is set to AUTO)

Oracle Database uses the INSTANCE_NUMBER parameter to distinguish among instances at startup. Oracle Database uses the INSTANCE_NAME parameter to assign redo log groups to specific instances. To simplify administration, use the same number for both the INSTANCE_NAME and INSTANCE_NUMBER parameters. If you have configured Grid Plug and Play, then the instance name parameter is optional and not required. If you do not specify the instance name, then instance name defaults to *db_unique_name_number*.

> **See Also:**
>
> - *Oracle Database 2 Day DBA*
> - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in Oracle RAC environments

Parameters that Should Have Identical Settings on All Instances

Oracle recommends that you set the values for the following parameters to the same value on all instances. Although you can have different settings for these parameters on different instances, setting each parameter to the same value on all instances simplifies administration.

Parameter	Description
ARCHIVE_LAG_TARGET	Different values for instances in your Oracle RAC database are likely to increase overhead because of additional automatic synchronization performed by the database processing. When using Oracle Streams with your Oracle RAC database, the value should be greater than zero.
CLUSTER_DATABASE_INSTANCES	While it is preferable for this parameter to have identical settings across all Oracle RAC database instances, it is not required.
LICENSE_MAX_USERS	This parameter determines a database-wide limit on the number of user accounts defined in the database. It is useful to have the same value on all instances of your database so you can see the current value no matter which instance you are using. Setting different values may cause Oracle Database to generate additional warning messages during instance startup, or cause commands related to database user management to fail on some instances.
LOG_ARCHIVE_FORMAT	If you do not use the same value for all your instances, then you unnecessarily complicate media recovery. The recovering instance expects the required archived redo log file names to have the format defined by its own value of LOG_ARCHIVE_FORMAT, regardless of which instance created the archived redo log files. Databases that support Oracle Data Guard, either to send or receive archived redo log files, must use the same value of LOG_ARCHIVE_FORMAT for all instances.
REDO_TRANSPORT_USER	This parameter specifies the name of the user whose password verifier is used when a remote login password file is used for redo transport authentication. This parameter is used in Oracle Data Guard configurations.
SPFILE	If this parameter does not identify the same file to all instances, then each instance may behave differently and unpredictably in fail over, load-balancing, and during normal operations. Additionally, a change you make to the SPFILE with an ALTER SYSTEM SET or ALTER SYSTEM RESET command is saved only in the SPFILE used by the instance where you run the command. Your change is not reflected in instances using different SPFILEs. If the SPFILE values are different in instances for which the values were set by the server, then you should restart the instances that are not using the default SPFILE.
TRACE_ENABLED	If you want diagnostic trace information to be always available for your Oracle RAC database, you must set TRACE_ENABLED to TRUE on all of your database instances. If you trace on only some of your instances, then diagnostic information might not be available when required should the only accessible instances be those with TRACE_ENABLED set to FALSE.
UNDO_RETENTION	By setting different values for UNDO_RETENTION in each instance, you are likely to reduce scalability and encounter unpredictable behavior following a failover. Therefore, you should carefully consider whether there are any benefits before you assign different values for this parameter to the instances in your Oracle RAC database.

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in Oracle RAC environments

About Modifying the SERVICE_NAMES Parameter for Oracle RAC

The `SERVICE_NAMES` initialization parameter specifies one or more names by which clients can connect to the instance. The instance registers its service names with the listener. When a client requests a service, the listener determines which instances offer the requested service and routes the client to the appropriate instance.

In an Oracle RAC database, you should *not* modify this parameter directly. Instead, define services for your database and database instances using the Clustered Managed Database Services page in Oracle Enterprise Manager. If you must change a service, then you can use either Enterprise Manager or SRVCTL. When you use either Oracle Enterprise Manager or SRVCTL to create and start the service, the `SERVICE_NAMES` parameter is updated automatically once the service is active.

See Also:

- "About Oracle Services" on page 6-2

About the Server Parameter File for Oracle Real Application Clusters

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

All instances in the cluster database use the same SPFILE at startup. Oracle RAC uses a traditional parameter file only if an SPFILE does not exist or if you specify `PFILE` in your `STARTUP` command. Oracle recommends that you use an SPFILE to simplify administration, maintain parameter setting consistency, and to guarantee parameter setting persistence across database shutdown and startup events. In addition, you can configure RMAN to back up your SPFILE.

See Also:

- *Oracle Database 2 Day DBA* for information about viewing and modifying the initialization parameter settings for your database
- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about using a server parameter file in an Oracle Real Application Clusters environment
- *Oracle Database Backup and Recovery User's Guide* for information about using RMAN to backup the SPFILE

Editing Initialization Parameter Settings for an Oracle RAC Database

You can use Enterprise Manager to view and edit the initialization parameter settings for your Oracle RAC database.

To view or modify the initialization parameters using Enterprise Manager:

1. Access the Cluster Database Home page.

See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Administration** menu, select **Initialization Parameters**.

   If the Database Login page appears, then log in as a SYSDBA user, for example, SYS or SYSTEM.

3. In the Initialization Parameters page, select either the **Current** or **SPFile** subpage to modify the parameter settings.

   See either:

   - Modifying the Initialization Parameter for Oracle RAC Using the Current Tab in Oracle Enterprise Manager
   - Modifying the Initialization Parameter for Oracle RAC Using the SPFile Tab in Oracle Enterprise Manager

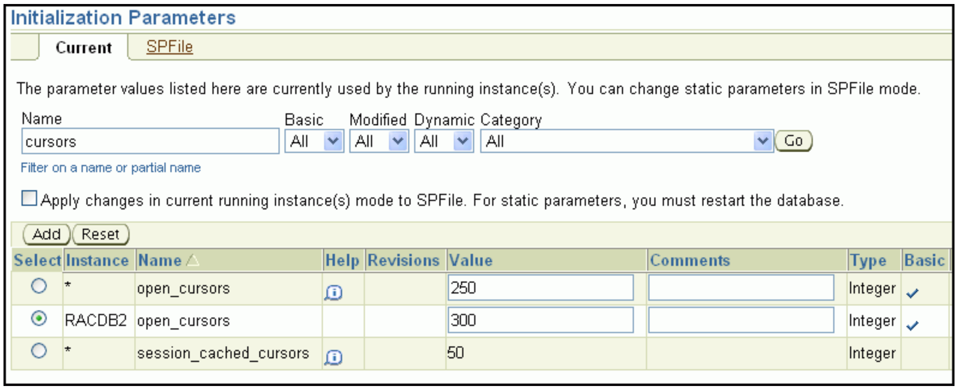
   See Also:

   - *Oracle Database 2 Day DBA*
   - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in Oracle RAC environments

Modifying the Initialization Parameter for Oracle RAC Using the Current Tab in Oracle Enterprise Manager

The Current subpage of the Initialization Parameters page contains a list of configuration parameters for that instance and database. You can set these parameters to particular values to initialize many of the memory and process settings of an Oracle instance. When you modify initialization parameters using the Current tab, the changes are applied only to the running instances, not the SPFILE, unless the "Apply changes in current running instance(s) mode to SPFile" option is selected.

The Instance column shows the instances for which the parameter has the value listed in the table. An asterisk (*) indicates that the parameter has the same value for all remaining instances of the cluster database. For example, if `open_cursors = 250` for `node1` and `open_cursors = 300` for `node2`, then the Instance column for `open_cursors = 250` displays an asterisk, but the Instance column for `open_cursors = 300` contains "node2". This shorthand notation saves space when the cluster database has many instances.



Initialization Parameters

The parameter values listed here are currently used by the running instance(s). You can change static parameters in SPFile mode.

Name	Basic	Modified	Dynamic	Category
cursors	All	All	All	All

Filter on a name or partial name

□ Apply changes in current running instance(s) mode to SPFile. For static parameters, you must restart the database.

Select	Instance	Name	Help	Revisions	Value	Comments	Type	Basic
○	*	open_cursors	ⓘ		250		Integer	✓
●	RACDB2	open_cursors			300		Integer	✓
○	*	session_cached_cursors	ⓘ		50		Integer	

You can filter the Initialization Parameters page to show only those parameters that meet the criteria of the filter you enter in the Filter by name field. Optionally, you can select **Show All** to display on one page all parameters currently used by the running instance(s).

To modify parameter values using the Current tab:

1. Select a parameter from the Select column and do one of the following:

   - Click **Add** to add the selected parameter to a different instance.

     When the new row appears in the table, choose the instance name to which the new setting applies, and enter the new value for the parameter in the Value field. Optionally, you can put text in the Comment field to indicate why the instance has a different value from the other instances.

   - Click **Reset** to reset the value of the selected parameter. Note that using this option you can reset only those parameters that do not have an asterisk in the Instance column.

     When you click Reset, the value of the selected parameter is reset to the value of the remaining instances (that is, the row with the asterisk in the Instance column) and the row that has the local instance name in the Instance field is removed from the table.

   For example, select the parameter `OPEN_CURSORS`, then click **Add**. In the new entry for `OPEN_CURSORS`, select an **Instance** (for example `sales1` or `RACDB2`), and change the **Value** field to 300.

2. After you make changes to one or more of the parameters, click **Apply** to accept and implement the changes.

See Also:

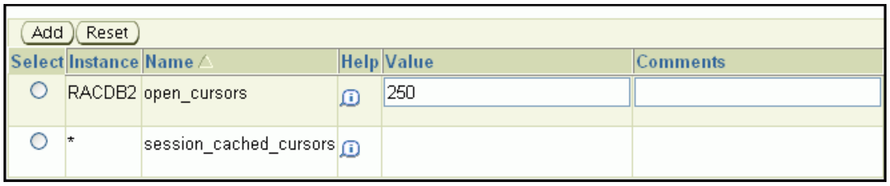
- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in Oracle RAC environments

Modifying the Initialization Parameter for Oracle RAC Using the SPFile Tab in Oracle Enterprise Manager

You can also add or reset parameters using the SPFile tab. When you modify initialization parameters using the SPFile tab, the changes are applied only to the SPFILE, not the currently running instances, unless the "Apply changes in SPFile mode to the current running instance(s)" option is selected.

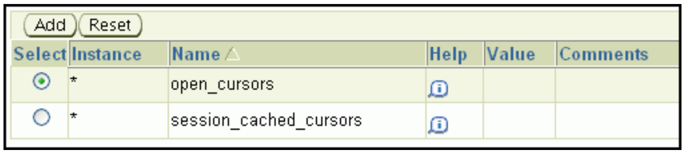
Resetting parameters using the SPFile tab is different than resetting the same parameters using the Current tab. You can either reset the parameter value for an instance back to the default value for all instances, or you can delete the default parameter setting (unset the parameter) for all instances.

- If you reset a parameter with an asterisk in the Instance column, then Oracle deletes the entry from both the SPFILE and the table displayed in Enterprise Manager. Only parameters without an asterisk (or instance-specific parameters) remain.



Select	Instance	Name △	Help	Value	Comments
○	RACDB2	open_cursors	ⓘ	250	
○	*	session_cached_cursors	ⓘ		

- If you reset the only entry for a nonasterisk parameter, then Oracle deletes the parameter from both the SPFILE and the table in Enterprise Manager, but Oracle replaces the parameter by a *dummy* parameter with an empty value field and an asterisk in the Instance column. Using a dummy parameter enables you to specify a new value for the parameter, add new instance-specific entries for the parameter, and so on.



Select	Instance	Name △	Help	Value	Comments
●	*	open_cursors	ⓘ		
○	*	session_cached_cursors	ⓘ		

Resetting a parameter that is set for only one instance results in the parameter being unset for all instances.

To modify parameter values using the SPFile tab:

1. Choose a parameter using the **Select** column.

   For more information about each parameter, click the information icon in the Help column next to the parameter.

2. Click **Add**, then edit the entry to change the target instance or parameter value. You can optionally add a comment describing the reason for the change.

3. Click **Apply** to apply the changes to the SPFILE, or to the SPFile and the currently running instance if you selected the "Apply changes in SPFile mode to the current running instance(s)" option.

4. On the SPFile subpage of the Initialization Parameters page, select a parameter from the Select column and do one of the following:

   - Click **Add** to add the selected parameter to a different instance.

     When the new row appears in the table, choose the instance name to which the new setting applies, and then enter the new value for the parameter in the Value field. Optionally, you can put text in the Comment field describing the reason for the change.

   - Click **Reset** to reset the value of the selected parameter.

     When you click Reset, one of the following actions is performed:

     – If the entry you selected was for a specific instance, then the value of the selected parameter for that instance is reset to the value of the remaining instances (indicated by an asterisk in the Instance column). The entry that has the local instance name in the Instance field is deleted.

     – If the entry you selected to reset was the default value for all instances (indicated by an asterisk in the Instance column), then the value of the selected parameter is unset for all instances, but any instance-specific parameter entries for the same parameter are not changed.

  - If you reset the only entry for a parameter, regardless of whether the entry applies to all instances or a single instance, then the parameter is unset for all instances in the cluster database.

5. After you make changes to one or more of the parameters, click **Apply** to accept and implement the changes.

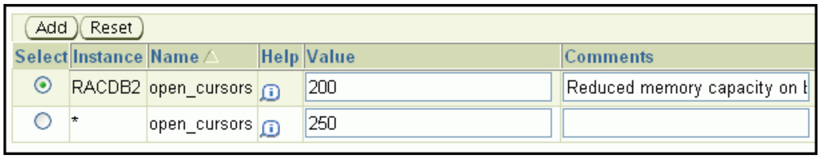
See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in Oracle RAC environments

Example: Modifying the OPEN_CURSORS Parameter

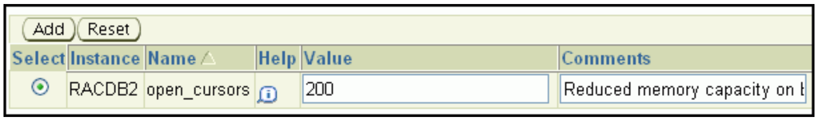
Suppose that the `open_cursors` parameter has two entries in the SPFILE:

```
*.open_cursors = 200
RACDB2.open_cursors = 250
```



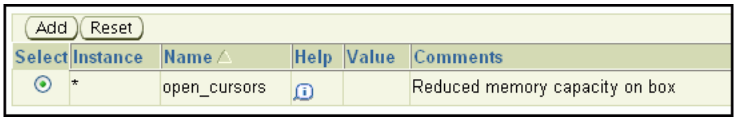
Select	Instance	Name △	Help	Value	Comments
●	RACDB2	open_cursors	ⓘ	200	Reduced memory capacity on
○	*	open_cursors	ⓘ	250	

Using the Initialization Parameters page with the SPFile tab selected, if you click **Reset** for `*.open_cursors`, then Enterprise Manager deletes that entry from both the SPFILE and the displayed list of parameters, leaving only `RACDB2.open_cursors = 250` displayed.



Select	Instance	Name △	Help	Value	Comments
●	RACDB2	open_cursors	ⓘ	200	Reduced memory capacity on

If you click **Reset** for `RACDB2.open_cursors`, then Enterprise Manager also deletes this parameter entry from both the SPFILE and the displayed list of parameters, but then a new entry, `*.open_cursors = <NULL>` is added to the displayed list of parameters for the reset parameter.



Select	Instance	Name △	Help	Value	Comments
●	*	open_cursors	ⓘ		Reduced memory capacity on box

See Also:

- *Oracle Database 2 Day DBA* for information about viewing and modifying the initialization parameter settings for your database
- *Oracle Database 2 Day DBA*
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about using a server parameter file in an Oracle Real Application Clusters environment

About Administering Storage in Oracle RAC

Most administration tasks for managing storage are the same for Oracle single-instance and Oracle RAC databases. This section provides additional information for using Enterprise Manager to manage some storage structures of an Oracle RAC database.

This section describes the following topics:

- About Automatic Undo Management in Oracle RAC
- Oracle Automatic Storage Management in Oracle RAC
- Administering Redo Logs in Oracle RAC

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*

About Automatic Undo Management in Oracle RAC

Oracle RAC automatically manages undo segments within a specific undo tablespace that is assigned to an instance. Only the instance assigned to the undo tablespace can modify the contents of that tablespace. However, each instance can read the undo data blocks created by any instance. Also, when performing transaction recovery, any instance can update any undo tablespace, if that undo tablespace is not currently being used by another instance for undo generation or transaction recovery.

You assign undo tablespaces in your Oracle RAC database by specifying a different value for the `UNDO_TABLESPACE` parameter for each instance in your SPFILE or individual PFILEs. You cannot simultaneously use automatic undo management and manual undo management in an Oracle RAC database. In other words, all instances of an Oracle RAC database must operate in the same undo mode.

See Also:

- *Oracle Database 2 Day DBA* for more information about managing the undo data for your database

Oracle Automatic Storage Management in Oracle RAC

Oracle Automatic Storage Management (Oracle ASM) automatically optimizes storage to maximize performance by managing the storage configuration across the disks that Oracle ASM manages. Oracle ASM does this by evenly distributing the storage load across all the available storage within your cluster database environment. Oracle ASM partitions your total disk space requirements into uniformly sized units across all the disks in a disk group. Oracle ASM can also automatically mirror data to prevent data loss. Because of these features, Oracle ASM also significantly reduces your administrative overhead.

As in single-instance Oracle databases, using Oracle ASM in Oracle RAC does not require I/O tuning. The following topics describe Oracle ASM and Oracle ASM administration:

- About Oracle ASM Components in Oracle RAC
- About Disk Group Configurations for Oracle ASM in Oracle RAC
- About Standalone Oracle ASM Disk Group Management
- About Oracle ASM Instance and Disk Group Management

See Also:

- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*

About Oracle ASM Components in Oracle RAC

When you configure Oracle ASM after installing Oracle Grid Infrastructure for a cluster, one Oracle ASM instance is created on each node in your Oracle RAC environment if one does not exist. Each Oracle ASM instance has either an SPFILE or PFILE type parameter file. For the environment described in this guide, the Oracle ASM instances use SPFILEs.

See Also: "About Oracle Automatic Storage Management"

About Disk Group Configurations for Oracle ASM in Oracle RAC

When you create a disk group for a cluster, or add new disks to an existing clustered disk group, you must prepare only the underlying physical storage on shared disks. The shared disk requirement is the only substantial difference between using Oracle ASM in an Oracle RAC database compared to using it in a single-instance Oracle database. Oracle ASM automatically rebalances the storage load after you add or delete a disk or disk group.

In a cluster, each Oracle ASM instance manages the metadata updates to the disk groups for the node on which it is running. In addition, each Oracle ASM instance coordinates disk group metadata with other nodes in the cluster. See Figure 1–2, "Oracle Instances and Oracle ASM Storage" for an illustration of the relationship between Oracle RAC instances, Oracle ASM instances, and the shared storage. As in single-instance Oracle databases, you can use Enterprise Manager, Oracle ASM configuration assistant (ASMCA), Oracle ASM command-line utility (ASMCMD), SQL*Plus, and SRVCTL to administer disk groups for Oracle ASM in an Oracle RAC environment.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about how to use SQL*Plus to administer Oracle ASM instances

About Standalone Oracle ASM Disk Group Management

You can configure Oracle ASM during installation of Oracle Grid Infrastructure for a cluster. If you choose not to configure Oracle ASM during installation, then you must manually configure Oracle ASM, or perform standalone Oracle ASM disk group management. You do not have to create a database to modify Oracle ASM storage properties.

To manually configure Oracle ASM, you must first run ASMCA to create the Oracle ASM instances and create at least one disk group. After Oracle ASM is running, then you can use DBCA to create an Oracle RAC database that uses Oracle ASM for storage.

Note: If you installed the Oracle Cluster Registry (OCR) and Voting Disks on Oracle ASM as part of your Oracle Grid Infrastructure for a cluster install, then OUI creates the Oracle ASM instances and you do not have to run ASMCA. You must use ASMCA only if you did not specify Oracle ASM storage for the OCR and Voting disks during installation.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about how to use the Oracle Automatic Storage Management command-line utility

About Oracle ASM Instance and Disk Group Management

You can perform administrative operations on Oracle ASM disk groups using Enterprise Manager such as adding and deleting them. You can also monitor Oracle ASM disk group performance and control disk group availability at the instance level. For example, Enterprise Manager provides some Oracle RAC-specific features for Oracle ASM, such as the following:

- When you add a disk group, the disk group definition includes a check box to indicate whether the disk group is automatically mounted to all the cluster database instances.
- The default Disk Group Performance page displays instance-level performance details when you click a performance characteristic such as **Write Response Time** or **I/O Throughput**.
- When you mount and dismount Oracle ASM disk groups, you can use a check box to indicate which instances should mount or dismount a particular Oracle ASM Disk Group.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for more information about managing disk groups

Administering Redo Logs in Oracle RAC

Managing redo log files in Oracle RAC environments is similar to managing redo log files in single-instance Oracle Database environments. The following sections provide an overview of additional concepts and procedures for configuring redo log files in Oracle RAC environments:

- About Redo Log Groups and Redo Threads in Oracle RAC Databases
- About Accessing Redo Log Files for an Oracle RAC Database
- Using Enterprise Manager to View and Create Online Redo Log Files

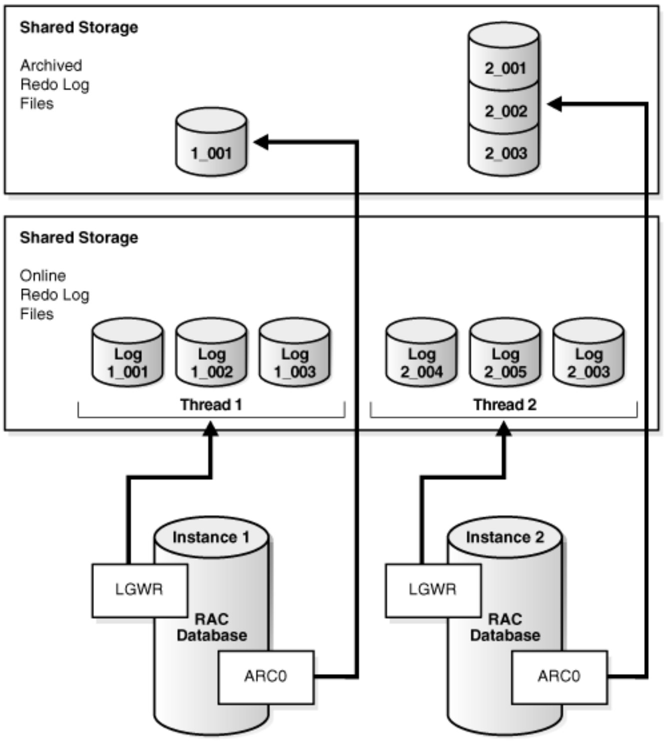
See Also:

- *Oracle Database 2 Day DBA* for more information about online redo logs
- *Oracle Automatic Storage Management Administrator's Guide* for informattion about creating online redo logs groups in Oracle ASM

About Redo Log Groups and Redo Threads in Oracle RAC Databases

Redo logs contain a record of changes that have been made to data files. In a single-instance Oracle database, redo logs are stored in two or more redo log file groups. Each of these groups contains a redo log file and possibly one or more mirrored copies of that file. In an Oracle RAC database, each instance requires its own set of redo log groups, which is known as a **redo thread**. Mirrored copies of the redo log files provide your system with extra protection against data loss that is due to hardware failures or data corruption. If a redo log file is unreadable, then Oracle Database attempts to access its mirrored copy. The redo log file mirrors should be located on different disk devices from the primary redo log files.

Figure 4–1 Illustration of Redo Threads for an Oracle RAC Database



Each instance's redo thread must contain at least two redo log groups. Each redo log group should contain at least two members: a redo log and its mirrored copy. If you create your Oracle RAC database using DBCA, then your Oracle RAC database automatically implements a configuration that meets the Oracle recommendations.

You should create redo log groups only if you are using administrator-managed databases. For policy-managed databases, if an instance starts due to a change in server pool cardinality, then Oracle Database automatically creates redo log files, enables a redo thread for the instance if there is not a redo thread allocated to that instance, and creates the undo tablespace if there is not an undo tablespace allocated to that instance. The database must be using Oracle Managed Files and Oracle ASM in this situation. See *Oracle Real Application Clusters Administration and Deployment Guide* for more information about redo log file storage in Oracle RAC.

In an Oracle RAC database, all the redo log files reside on shared storage. In addition, each instance must have access to the redo log files of all the other instances in the cluster. If your Oracle RAC database uses Oracle ASM, then Oracle ASM manages the shared storage for the redo log files and the access to those files.

Note: DBCA creates redo threads and undo tablespaces up to the maximum value specified by the `CLUSTER_DATABASE_INSTANCES` parameter at the time of creation for better performance.

See Also:

- *Oracle Database 2 Day DBA* for more information about online redo logs
- *Oracle Automatic Storage Management Administrator's Guide* for informattion about creating online redo logs groups in Oracle ASM

About Accessing Redo Log Files for an Oracle RAC Database

In an Oracle RAC database, each instance writes and archives the redo log groups in its redo thread in the same manner that single-instance Oracle databases do. However, in recovery mode, the instance performing the recovery can read and process all the redo threads for the database, regardless of which instance generated the redo thread. Being able to read all the redo threads enables a running instance to recover the work completed by one or more failed instances.

In case of instance failure, a surviving instance can read the redo logs of the failed instance. Users can continue to access and update the database without waiting for the failed instance to be restarted. For example, assume that you have an Oracle RAC database with two instances, instance A and instance B. If instance A is down, then instance B can read the redo log files for both instance A and B to ensure a successful recovery.

See Also: "Administering Redo Logs in Oracle RAC" on page 4-15

Using Enterprise Manager to View and Create Online Redo Log Files

On the Redo Log Groups page, you can create additional redo log groups and add members to the redo log group. The Thread column identifies the instance, or redo thread, to which a redo log file belongs.

To access the redo log file groups using Enterprise Manager:

1. Access the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Administration** menu, select **Storage**, and then select **Redo Log Groups**.

   The Redo Log Groups page appears.

3. On this page you can perform the following tasks:

   - View the status, number of members, thread, file size, status, and archive status of each redo log group
   - Create or delete log groups
   - Edit a redo log group to add or remove members
   - Perform other redo log group management tasks, including clearing log files, duplicating a log group, generating sizing advice, and forcing a log switch

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for additional information about redo threads in an Oracle RAC environment
- *Oracle Automatic Storage Management Administrator's Guide* for informattion about creating online redo logs groups in Oracle ASM
- *Oracle Database 2 Day DBA* for more information about managing online redo log files

5 Administering Oracle Clusterware Components

The main components to manage in your Oracle Clusterware environment are the voting disks and the Oracle Cluster Registry (OCR).

- About Oracle Clusterware
- Managing the Oracle Clusterware Stack
- Administering Voting Disks for Oracle Clusterware
- Backing Up and Recovering the Oracle Cluster Registry
- Changing the Oracle Cluster Registry Configuration
- Troubleshooting the Oracle Cluster Registry

See Also:

- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "About Verifying the Oracle Clusterware Installation"

About Oracle Clusterware

Oracle Real Application Clusters (Oracle RAC) uses Oracle Clusterware as the infrastructure that binds multiple nodes that then operate as a single server. In an Oracle RAC environment, Oracle Clusterware monitors all Oracle components (such as instances and listeners). If a failure occurs, then Oracle Clusterware automatically attempts to restart the failed component and also redirects operations to a surviving component.

Oracle Clusterware includes a high availability framework for managing any application that runs on your cluster. Oracle Clusterware manages applications to ensure they start when the system starts. Oracle Clusterware also monitors the applications to ensure they are always available. For example, if an application process fails, then Oracle Clusterware attempts to restart the process based on scripts that you customize. If a node in the cluster fails, then you can program application processes that typically run on the failed node to restart on another node in the cluster.

See Also:

- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "Using Oracle Universal Installer to Install the Oracle Grid Infrastructure for a Cluster"

This section contains the following topics:

- About the Voting Disks
- About Oracle Cluster Registry
- About High Availability of Oracle Clusterware Files

About the Voting Disks

The **voting disk** records node membership information. A node must be able to access more than half the voting disks at any time. To avoid simultaneous loss of multiple voting disks, each voting disk should be on a storage device that does not share any components (controller, interconnect, and so on) with the storage devices used for the other voting disks.

For example, if you have five voting disks configured, then a node must be able to access at least three of the voting disks at any time. If a node cannot access the minimum required number of voting disks, then it is *evicted*, or removed, from the cluster. After the cause of the failure has been corrected and access to the voting disks has been restored, you can instruct Oracle Clusterware to recover the failed node and restore it to the cluster.

About Oracle Cluster Registry

Oracle Cluster Registry (OCR) is a file that contains information about the cluster node list and instance-to-node mapping information. OCR also contains information about Oracle Clusterware resource profiles for resources that you have customized. The voting disk data is also backed up in OCR.

Each node in a cluster also has a local copy of the OCR, called an Oracle Local Registry (OLR), that is created when Oracle Clusterware is installed. Multiple processes on each node have simultaneous read and write access to the OLR particular to the node on which they reside, whether Oracle Clusterware is fully functional. By default, OLR is located at `Grid_home/cdata/$HOSTNAME.olr`

About High Availability of Oracle Clusterware Files

High availability configurations have redundant hardware and software that maintain operations by avoiding single points of failure. When a component is down, Oracle Clusterware redirects its managed resources to a redundant component. However, if a disaster strikes, or a massive hardware failure occurs, then having redundant components might not be enough. To fully protect your system it is important to have backups of your critical files.

The Oracle Clusterware installation process creates the voting disk and the OCR on shared storage. If you select the option for normal redundant copies during the installation process, then Oracle Clusterware automatically maintains redundant copies of these files to prevent the files from becoming single points of failure. The normal redundancy feature also eliminates the need for third-party storage redundancy solutions. When you use normal redundancy, Oracle Clusterware automatically maintains two copies of the OCR file and three copies of the voting disk file.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

Managing the Oracle Clusterware Stack

By default, Oracle Clusterware is configured to restart whenever the server it resides on is restarted. During certain maintenance operations, you may be required to stop or start the Oracle Clusterware stack manually.

This section contains the following topics:

- Starting Oracle Clusterware
- Stopping Oracle Clusterware

Note: Do not use Oracle Clusterware Control (CRSCTL) commands on Oracle entities (such as resources, resource types, and server pools) that have names beginning with `ora` unless you are directed to do so by Oracle Support. The Server Control utility (SRVCTL) is the correct utility to use on Oracle entities.

See Also:

- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "About Oracle Clusterware"

Starting Oracle Clusterware

You use the CRSCTL utility to manage Oracle Clusterware. If the Oracle High Availability Services daemon (OHASD) is running on all the cluster nodes, then you can start the entire Oracle Clusterware stack (all the processes and resources managed by Oracle Clusterware) on all nodes in the cluster by executing the following command on any node:

```
crsctl start cluster -all
```

You can start the Oracle Clusterware stack on specific nodes by using the `-n` option followed by a space-delimited list of node names, for example:

```
crsctl start cluster -n racnode1 racnode4
```

To use the previous command, the OHASD process must be running on the specified nodes.

To start the entire Oracle Clusterware stack on a node, including the OHASD process, run the following command on that node:

```
crsctl start crs
```

See Also:

- "About Oracle Clusterware"
- "Managing the Oracle Clusterware Stack"

Stopping Oracle Clusterware

To stop Oracle Clusterware on all nodes in the cluster, execute the following command on any node:

```
crsctl stop cluster -all
```

The previous command stops the resources managed by Oracle Clusterware, the Oracle ASM instance, and all the Oracle Clusterware processes (except for OHASD and its dependent processes).

To stop Oracle Clusterware and Oracle ASM on select nodes, include the `-n` option followed by a space-delimited list of node names, for example:

```
crsctl stop cluster -n racnode1 racnode3
```

If you do not include either the `-all` or the `-n` option in the `stop cluster` command, then Oracle Clusterware and its managed resources are stopped only on the node where you execute the command.

To completely shut down the entire Oracle Clusterware stack, including the OHASD process, use the `crsctl stop crs` command. CRSCTL attempts to gracefully stop the resources managed by Oracle Clusterware during the shutdown of the Oracle Clusterware stack. If any resources that Oracle Clusterware manages are still running after executing the `crsctl stop crs` command, then the command fails. You must then use the `-f` option to unconditionally stop all resources and stop the Oracle Clusterware stack, for example:

```
crsctl stop crs -f
```

Note: When you shut down the Oracle Clusterware stack, you also shut down the Oracle Automatic Storage Management (Oracle ASM) instances. If the Oracle Clusterware files (voting disk and OCR) are stored in an Oracle ASM disk group, then the only way to shut down the Oracle ASM instances is to shut down the Oracle Clusterware stack.

See Also:

- "About Oracle Clusterware"
- "Managing the Oracle Clusterware Stack"

Administering Voting Disks for Oracle Clusterware

This section describes how to perform the following tasks:

- Adding and Removing Voting Disks
- Backing Up and Recovering Voting Disks
- Migrating Voting Disks to Oracle ASM Storage

Adding and Removing Voting Disks

If you choose to store Oracle Clusterware files on Oracle ASM and use redundancy for the disk group, then Oracle ASM automatically maintains the ideal number of voting files based on the redundancy of the disk group.

If you use a different form of shared storage to store the voting disks, then you can dynamically add and remove voting disks after installing Oracle RAC. Do this using the following commands where *`path`* is the fully qualified path for the additional voting disk.

To add or remove a voting disk that is stored on disk:

1. Run the following command as the `grid` user to add a voting disk:

   ```
   crsctl add css votedisk path
   ```

2. Run the following command as the `grid` user to remove a voting disk:

   ```
   crsctl delete css votedisk path
   ```

> **See Also:**
>
> - "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
> - "Backing Up and Recovering Voting Disks"

Backing Up and Recovering Voting Disks

This section contains the following topics:

- Backing Up Voting Disks
- Replacing Voting Disks
- Restoring Voting Disks

> **See Also:**
>
> - "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
> - "About Oracle Clusterware"
> - "Performing Backups of Your Oracle Real Application Clusters Database"
> - *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

Backing Up Voting Disks

The voting disk data is automatically backed up in OCR as part of any configuration change so you do not have to perform manual backups of the voting disk. The voting disk files are backed up automatically by Oracle Clusterware if the contents of the files have changed in the following ways:

- Configuration parameters, for example `misscount`, have been added or modified
- After performing voting disk `add` or `delete` operations

> **See Also:**
>
> - "Backing Up and Recovering Voting Disks"
> - "About Oracle Clusterware"
> - *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

Replacing Voting Disks

If a voting disk is damaged, and no longer usable by Oracle Clusterware, then you can replace or re-create the voting disk. You replace a voting disk by deleting the unusable voting disk and then adding a new voting disk to your configuration. The voting disk contents are restored from a backup when a new voting file is added; this occurs regardless of whether the voting disk file is stored in Oracle Automatic Storage Management (Oracle ASM).

To replace a corrupt, damaged, or missing voting disk that is not stored in Oracle ASM:

1. Use CRSCTL to remove the damaged voting disk. For example, if the damaged voting disk is stored in the disk location `/dev/sda3`, then execute the command:

   ```
   crsctl delete css votedisk /dev/sda3
   ```

2. Use CRSCTL to create a new voting disk in the same location, for example:

   ```
   crsctl add css votedisk /dev/sda3
   ```

See Also:

- "Backing Up and Recovering Voting Disks"
- "About Oracle Clusterware"
- *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

Restoring Voting Disks

If all voting disks are corrupted, then you can restore them as described in *Oracle Clusterware Administration and Deployment Guide*.

> **Note:** Restoring a voting disk from a copy created with the Linux or UNIX operating system `dd` command is not supported

See Also:

- "Backing Up and Recovering Voting Disks"
- "About Oracle Clusterware"
- *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

Migrating Voting Disks to Oracle ASM Storage

You can store the Oracle Clusterware voting disk files in an Oracle ASM disk group. If you choose to store your voting disks in Oracle ASM, then Oracle ASM stores all the voting disks for the cluster in the disk group you choose. You cannot combine voting disks stored in Oracle ASM and voting disks not stored in Oracle ASM in the same cluster.

The number of voting files you can store in a particular Oracle ASM disk group depends upon the redundancy of the disk group. By default, Oracle ASM puts each voting disk in its own failure group within the disk group. A normal redundancy disk group must contain at least two failure groups but if you are storing your voting disks on Oracle ASM, then a normal redundancy disk group must contain at least three failure groups. A high redundancy disk group must contain at least three failure groups.

Once you configure voting disks on Oracle ASM, you can only make changes to the voting disks' configuration using the `crsctl replace votedisk` command. This is true even in cases where there are no working voting disks. Despite the fact that the `crsctl query css votedisk` command reports zero voting disks in use, Oracle Clusterware remembers the fact that Oracle ASM was in use and the `replace` verb is required. Only after you use the `replace` verb to move voting disks back to non-Oracle ASM storage are the CRSCTL commands `add css votedisk` and `delete css votedisk` again usable.

To move voting disks from shared storage to an Oracle ASM disk group:

1. Use the Oracle ASM Configuration Assistant (ASMCA) to create an Oracle ASM disk group.
2. Verify that the ASM Compatibility attribute for the disk group is set to 12.1.0.0 or higher.
3. Use CRSCTL to create a voting disk in the Oracle ASM disk group by specifying the disk group name in the following command:

```
crsctl replace votedisk +ASM_disk_group
```

See Also:

- "Creating Additional Oracle ASM Disk Groups" on page 3-11
- *Oracle Automatic Storage Management Administrator's Guide* for more information about disk group compatibility attributes
- *Oracle Clusterware Administration and Deployment Guide* for more information about managing voting disks

Backing Up and Recovering the Oracle Cluster Registry

Oracle Clusterware automatically creates OCR backups every four hours. At any one time, Oracle Clusterware always retains the latest three backup copies of the OCR that are four hours old, one day old, and one week old.

You cannot customize the backup frequencies or the number of files that Oracle Clusterware retains. You can use any backup software to copy the automatically generated backup files at least once daily to a different device from where the primary OCR file resides.

This section contains the following topics:

- Viewing Available OCR Backups
- Manually Backing Up the OCR
- Recovering the OCR

See Also:

- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "Changing the Oracle Cluster Registry Configuration"
- "Troubleshooting the Oracle Cluster Registry"

Viewing Available OCR Backups

Use the `ocrconfig` utility to view the backups generated automatically by Oracle Clusterware.

To find the most recent backup of the OCR:

Run the following command on any node in the cluster:

```
ocrconfig -showbackup
```

See Also:

- "Manually Backing Up the OCR"
- "Backing Up and Recovering the Oracle Cluster Registry"
- "Troubleshooting the Oracle Cluster Registry"

Manually Backing Up the OCR

Use the `ocrconfig` utility to force Oracle Clusterware to perform a backup of OCR at any time, rather than wait for the automatic backup that occurs at four-hour intervals. This option is especially useful when you want to obtain a binary backup on demand, such as before you make changes to OCR.

To manually backup the contents of the OCR:

1. Log in as the `root` user.
2. Use the following command to force Oracle Clusterware to perform an immediate backup of the OCR:

   ```
   ocrconfig -manualbackup
   ```

   The date and identifier of the recently generated OCR backup is displayed.
3. (Optional) If you must change the location for the OCR backup files, then use the following command, where *`directory_name`* is the new location for the backups:

   ```
   ocrconfig -backuploc directory_name
   ```

The default location for generating backups on Oracle Linux systems is *`Grid_home`*`/cdata/`*`cluster_name`* where *`cluster_name`* is the name of your cluster and *Grid_home* is the home directory of the Oracle Grid Infrastructure for a cluster installation. Because the default backup is on a local file system, Oracle recommends that you include the backup file created with the `ocrconfig` command as part of your operating system backup using standard operating system or third-party tools.

Note: You can use the `ocrconfig -backuploc` command to change the location where the OCR backups are created.

See Also:

- "Backing Up and Recovering the Oracle Cluster Registry"
- "Replacing an OCR"
- "Repairing an OCR Configuration on a Local Node"

Recovering the OCR

There are two methods for recovering the OCR. The first method uses automatically generated OCR file copies and the second method uses manually created OCR export files.

This section contains the following topics:

- Checking the Status of the OCR
- Restoring the OCR from Automatically Generated OCR Backups

See Also:

- "About the OCRCHECK Utility"
- "Repairing an OCR Configuration on a Local Node"
- "Replacing an OCR"
- "Backing Up and Recovering the Oracle Cluster Registry"

Checking the Status of the OCR

In event of a failure, before you attempt to restore the OCR, ensure that the OCR is unavailable.

To check the status of the OCR:

1. Run the following command:

   ```
   ocrcheck
   ```

2. If this command does not display the message `'Device/File integrity check succeeded'` for at least one copy of the OCR, then all copies of the OCR have failed. You must restore the OCR from a backup or OCR export.
3. If there is at least one copy of the OCR available, then you can use that copy to restore the other copies of the OCR.

   See Also:

   - "About the OCRCHECK Utility"
   - "Repairing an OCR Configuration on a Local Node"
   - "Replacing an OCR"
   - "Backing Up and Recovering the Oracle Cluster Registry"

Restoring the OCR from Automatically Generated OCR Backups

When restoring the OCR from automatically generated backups, you first have to determine which backup file to use for the recovery.

To restore the OCR from an automatically generated backup on an Oracle Linux system:

1. Log in as the `root` user.
2. Identify the available OCR backups using the `ocrconfig` command:

   ```
   [root]# ocrconfig -showbackup
   ```

3. Review the contents of the backup using the following `ocrdump` command, where *`file_name`* is the name of the OCR backup file for which the contents should be written out to the file ocr_dump_output_file:

   ```
   [root]# ocrdump ocr_dump_output_file -backupfile file_name
   ```

   If you do not specify an output file name, then the utility writes the OCR contents to a file named `OCRDUMPFILE` in the current directory.

4. As the root user, stop Oracle Clusterware on all the nodes in your cluster by executing the following command:

   ```
   [root]# crsctl stop cluster -all
   ```

5. As the root user, restore the OCR by applying an OCR backup file that you identified in Step 2 using the following command, where *file_name* is the name of the OCR to restore. Ensure that the OCR devices that you specify in the OCR configuration exist, and that these OCR devices are valid before running this command.

   ```
   [root]# ocrconfig -restore file_name
   ```

6. As the root user, restart Oracle Clusterware on all the nodes in your cluster by running the following command:

   ```
   [root]# crsctl start cluster -all
   ```

7. Use the Cluster Verification Utility (CVU) to verify the OCR integrity. Exit the root user account, and, as the software owner of the Oracle Grid Infrastructure for a cluster installation, run the following command, where the -n all argument retrieves a list of all the cluster nodes that are configured as part of your cluster:

   ```
   cluvfy comp ocr -n all [-verbose]
   ```

See Also:

- "Viewing Available OCR Backups"
- "Manually Backing Up the OCR"
- "Backing Up and Recovering the Oracle Cluster Registry"

Changing the Oracle Cluster Registry Configuration

This section describes how to administer the Oracle Clusterware Registry (OCR). The OCR contains information about the cluster node list, which instances run on which nodes, and information about Oracle Clusterware resource profiles for applications that have been modified to be managed by Oracle Clusterware.

This section contains the following topics:

- Adding an OCR Location
- Migrating the OCR to Oracle ASM Storage
- Replacing an OCR
- Removing an OCR
- Repairing an OCR Configuration on a Local Node

Note: The operations in this section affect the OCR for the entire cluster. However, the ocrconfig command cannot modify OCR configuration information for nodes that are shut down or for nodes on which Oracle Clusterware is not running. Avoid shutting down nodes while modifying the OCR using the ocrconfig command.

See Also:

- "About Oracle Clusterware"
- "Backing Up and Recovering the Oracle Cluster Registry"

Adding an OCR Location

Oracle Clusterware supports up to five OCR copies. You can add an OCR location after an upgrade or after completing the Oracle RAC installation. Additional OCR copies provide greater fault tolerance.

To add an OCR file:

As the `root` user, enter the following command to add a new OCR file:

```
[root]# ocrconfig -add new_ocr_file_name
```

This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

See Also:

- "About Oracle Clusterware"
- "Installing Oracle Grid Infrastructure for a Cluster"

Migrating the OCR to Oracle ASM Storage

You can store the OCR in an Oracle ASM disk group. By default, the OCR is configured to use Oracle ASM when you perform a new installation of Oracle Clusterware. However, if you upgrade from a previous release, then you can migrate OCR to reside on Oracle ASM, and take advantage of the improvements in managing Oracle Clusterware storage.

The OCR inherits the redundancy of the disk group. If you want high redundancy for the OCR, then you must create an Oracle ASM disk group with high redundancy. You should use a disk group with at least normal redundancy, unless you have an external mirroring solution. If you store the OCR in an Oracle ASM disk group, and the Oracle ASM instance fails on a node, then the OCR becomes unavailable only on that node. The failure of one Oracle ASM instance does not affect the availability of the entire cluster.

Oracle does not support storing the OCR on different storage types simultaneously, such as storing the OCR on both Oracle ASM and a shared file system, except during a migration. After you have migrated the OCR to Oracle ASM storage, you must delete the existing OCR files.

To move the OCR from shared storage to an Oracle ASM disk group:

1. Use the Oracle ASM Configuration Assistant (ASMCA) to create an Oracle ASM disk group that is at least the same size as the existing OCR and has at least normal redundancy.

2. Verify that the ASM Compatibility attribute for the disk group is set to 11.2.0.0 or higher.

3. Run the following OCRCONFIG command as the `root` user, specifying the Oracle ASM disk group name:

   ```
   # ocrconfig -add +ASM_disk_group
   ```

You can run this command more than once if you add multiple OCR locations. You can have up to five OCR locations. However, each successive run must point to a different disk group.

4. Remove the non-Oracle ASM storage locations by running the following command as the `root` user:

   ```
   # ocrconfig -delete old_storage_location
   ```

   You must run this command once for every shared storage location for the OCR that is not using Oracle ASM.

   See Also:

   - "Creating Additional Oracle ASM Disk Groups" on page 3-11
   - *Oracle Automatic Storage Management Administrator's Guide* for more information about disk group compatibility attributes
   - *Oracle Clusterware Administration and Deployment Guide* for more information about migrating the OCR to Oracle ASM

Replacing an OCR

If you must change the location of an existing OCR, or change the location of a failed OCR to the location of a working one, then you can use the following procedure if one OCR file remains online.

To change the location of an OCR or replace an OCR file:

1. Use the OCRCHECK utility to verify that a copy of the OCR other than the one you are going to replace is *online*, using the following command:

   ```
   ocrcheck
   ```

   Note: The OCR that you are *replacing* can be either online or offline.

2. Use the following command to verify that Oracle Clusterware is running on the node on which you are going to perform the replace operation:

   ```
   crsctl check cluster -all
   ```

3. As the `root` user, enter the following command to designate a new location for the specified OCR file:

   ```
   [root]# ocrconfig -replace source_ocr_file -replacement destination_ocr_file
   ```

   This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

4. Use the OCRCHECK utility to verify that OCR replacement file is online:

   ```
   ocrcheck
   ```

See Also:

- "About the OCRCHECK Utility"
- "Repairing an OCR Configuration on a Local Node"
- "Adding an OCR Location"

Removing an OCR

To remove an OCR file, at least one copy of the OCR must be online. You can remove an OCR location to reduce OCR-related overhead or to stop mirroring your OCR because you moved the OCR to a redundant storage system, such as a redundant array of independent disks (RAID).

To remove an OCR location from your cluster:

1. Use the OCRCHECK utility to ensure that at least one OCR other than the OCR that you are removing is online.

   ```
   ocrcheck
   ```

   > **Note:** Do *not* perform this OCR removal procedure unless there is at least one active OCR online.

2. As the `root` user, run the following command on any node in the cluster to remove a specific OCR file:

   ```
   [root]# ocrconfig -delete ocr_file_name
   ```

   This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

   See Also:

   - "About the OCRCHECK Utility"
   - "Replacing an OCR"
   - "Adding an OCR Location"

Repairing an OCR Configuration on a Local Node

If a node in your cluster was not available when you modified the OCR configuration, then you must repair the OCR configuration on that node before it is restarted.

To repair an OCR configuration:

1. As the `root` user, run one or more of the following commands on the node on which Oracle Clusterware is stopped, depending on the number and type of changes that were made to the OCR configuration:

   ```
   [root]# ocrconfig -repair -add new_ocr_file_name

   [root]# ocrconfig -repair -delete ocr_file_name

   [root]# ocrconfig -repair -replace source_ocr_file -replacement dest_ocr_file
   ```

   These commands update the OCR configuration only on the node from which you run the command.

Note: You *cannot* perform these operations on a node on which the Oracle Clusterware daemon is running.

2. Restart Oracle Clusterware on the node you have just repaired.
3. As the `root` user, check the OCR configuration integrity of your cluster using the following command:

```
[root]# ocrcheck
```

See Also:

- "About the OCRCHECK Utility"
- "Replacing an OCR"
- "Adding an OCR Location"

Troubleshooting the Oracle Cluster Registry

This section includes the following topics about troubleshooting the Oracle Cluster Registry (OCR):

- About the OCRCHECK Utility
- Common Oracle Cluster Registry Problems and Solutions

About the OCRCHECK Utility

The OCRCHECK utility displays the data block format version used by the OCR, the available space and used space in the OCR, the ID used for the OCR, and the locations you have configured for the OCR. The OCRCHECK utility calculates a checksum for all the data blocks in all the OCRs that you have configured to verify the integrity of each block. It also returns an individual status for each OCR file and a result for the overall OCR integrity check. The following is a sample of the OCRCHECK output:

```
Status of Oracle Cluster Registry is as follows :
   Version                  :          3
   Total space (kbytes)     :     262144
   Used space (kbytes)      :      16256
   Available space (kbytes) :     245888
   ID                       :  570929253
   Device/File Name         : +CRS_DATA
                              Device/File integrity check succeeded
...
                              Decive/File not configured

   Cluster registry integrity check succeeded

   Logical corruption check succeeded
```

The OCRCHECK utility creates a log file in the following directory, where *`Grid_home`* is the location of the Oracle Grid Infrastructure for a cluster installation, and *`hostname`* is the name of the local node:

```
Grid_home/log/hostname/client
```

The log files have names of the form `ocrcheck_`*`nnnnn`*`.log`, where *`nnnnn`* is the process ID of the operating session that issued the `ocrcheck` command.

See Also:

- "Backing Up and Recovering the Oracle Cluster Registry"
- "Viewing Available OCR Backups"

Common Oracle Cluster Registry Problems and Solutions

The following table describes common OCR problems and their corresponding solutions.

Table 5–1 Common OCR Problems and Solutions

Problem	Solution
The OCR is not mirrored.	Run the `ocrconfig` command with the `-add` option as described in the section "Adding an OCR Location" on page 5-11.
A copy of the OCR has failed and you must replace it. Error messages are being reported in Enterprise Manager or the OCR log file.	Run the `ocrconfig` command with the `-replace` option as described in the section "Replacing an OCR" on page 5-12.
OCRCHECK does not find a valid OCR, or all copies of the OCR are corrupted.	Run the `ocrconfig` command with the `-restore` option as described in the section "Restoring the OCR from Automatically Generated OCR Backups" on page 5-9.
The OCR configuration was updated incorrectly.	Run the `ocrconfig` command with the `-repair` option as described in the section "Repairing an OCR Configuration on a Local Node" on page 5-13.
You are experiencing a severe performance effect from updating multiple OCR files, or you want to remove an OCR file for other reasons.	Run the `ocrconfig` command with the `-delete` option as described in the section "Removing an OCR" on page 5-13.
You want to change the location or storage type currently used by the OCR.	Run the `ocrconfig` command with the `-replace` option while Oracle Clusterware is running, as described in the section "Replacing an OCR" on page 5-12. If some cluster nodes are down when you move the OCR, then you must run `ocrconfig -repair` on each node that was down before you start Oracle Clusterware on that node.

See Also:

- "About Verifying the Oracle Clusterware Installation"
- "Replacing an OCR"
- "Adding an OCR Location"
- "Repairing an OCR Configuration on a Local Node"

6

Managing Workloads Using Dynamic Database Services

Using dynamic database services, you can distribute the workload across database instances to achieve optimal database and cluster performance for users and applications, as described in the following topics:

- About Workload Management with Dynamic Database Services
- Creating Services
- Administering Services
- Configuring Clients for High Availability

About Workload Management with Dynamic Database Services

Applications using a clustered database generally want to load balance their workload across the cluster. Oracle Real Application Clusters (Oracle RAC) includes a highly available (HA) application framework that provides the necessary service and integration points between Oracle RAC and custom enterprise applications.

You can deploy Oracle RAC and single-instance Oracle database environments to use dynamic database service features in many different ways. Depending on the number of nodes and your environment's complexity and objectives, your choices for the optimal workload management and high availability configuration depend on a variety of considerations.

To implement dynamic database services for an Oracle RAC database, you can use several different features. This section contains the following topics:

- About Oracle Services
- About the Database Resource Manager
- About Oracle RAC High Availability Framework
- About Fast Application Notification (FAN)
- About FAN Callouts
- About Using Transaction Guard for Improving Client Failover
- About Application Continuity for Masking Outages
- About the Load Balancing Advisory
- About Connection Load Balancing
- About Run-time Connection Load Balancing

See Also:

- "Creating Services"
- "About Automatic Database Diagnostic Monitor and Oracle RAC Performance" on page 8-2

About Oracle Services

Dynamic database services, often referred to as simply services, are logical abstractions for managing workloads in Oracle Database. Services divide workloads into mutually disjoint groupings. Each service represents a workload with common attributes, service-level thresholds, and priorities.

A single service can represent an application, multiple applications or a subset of a single application. For example, the Oracle E-Business suite defines a service for each responsibility, such as general ledger, accounts receivable, order entry, and so on. A single service can be associated with one or more instances of an Oracle RAC database, and a single instance can support multiple services.

Note: A database service can only be offered on a single network.

Services provide the following benefits:

- Provide a single entity for managing applications that compete for the same resources
- Enable each workload to be managed as a unit
- Hide the complexity of the cluster from the client

To manage workloads, you can define services that you assign to a particular application or to a subset of an application's operations. You can also use services to manage the workload for different types of work. For example, online users can use one service while batch processing can use a different service and reporting can use yet another service type.

Traditionally an Oracle database provided a single service and all users connected to the same service. A database always has this default database service that is the database name. This service cannot be modified. It should only be used for administrative tasks because it always enables you to connect to the database. The default database service should not be used for high availability because it cannot be disabled or relocated. Always use user-defined database services for your applications.

Note: Do not use the default database service for application workloads because the default database service is for administration purposes and cannot be modified. The default database service has the same name as the DB_NAME or DB_UNIQUE_NAME database initialization parameter. Create at least one service as described in "Creating Services" on page 6-17.

When a user or application connects to a database, Oracle recommends that you use a service for the connection. Oracle Database automatically creates one database service when the database is created. For basic or administrative connections, this may be all you need. However, for more flexibility in the management of the applications connecting to the database and their workload, you should create one or more application services and specify which database instances offer the services.

You can define services for both policy-managed and administrator-managed databases.

- **Policy-managed database**: When you define services for a **policy-managed database**, you assign the service to a server pool where the database is running. You can define the service as either uniform (running on all instances in the server pool) or singleton (running on only one instance in the server pool).
- **Administrator-managed database**: When you define a service for an **administrator-managed database**, you define which instances normally support that service.These are known as the `PREFERRED` instances. You can also define other instances to support a service if the preferred instance fails. These are known as `AVAILABLE` instances. A service that runs on an administrator-managed database must have at least one `PREFERRED` instance.

Services are integrated with the Database Resource Manager, which enables you to restrict the resources that a service uses within an instance. In addition, Oracle Scheduler jobs can run using a service, as opposed to using a specific instance.

See Also:

- "Creating Services" on page 6-17
- "Administering Services" on page 6-21
- "About Workload Management with Dynamic Database Services" on page 6-1
- *Oracle Database 2 Day DBA* for more information about client connections to the database
- *Oracle Database Administrator's Guide* for more information about managing application workloads with database services

About Service Failover in Administrator-Managed Databases

During normal operation a service can be running on any combination of the preferred and available instances, up to the cardinality of the service (the number of `PREFERRED` instances defined). Only at initial startup of the service will Oracle Clusterware attempt to start the service on the `PREFERRED` instances. If an instance fails, and one of the instances in the combined list of preferred and available instances is not offering the service, the service will relocate. You can also manually relocate the service to one of the instances in the combined list of preferred and available instances not offering the service.

If a service fails over to an available instance, then the service is not moved back to its preferred instance automatically. However, you can automate the relocation of a service to its preferred instance by using a callout. For more information about callouts, see "About FAN Callouts" on page 6-7. An example of a callout script can be found in the section "Oracle RAC FAN Callout Start Services Script" in the following white paper, available on Oracle Technology Network:
`http://www.oracle.com/us/products/database/maa-wp-beehive-maa-130890.pdf`

If you configure a preferred instance for a service, but do not specify at least one available instance for the service, then the service does not relocate to another instance if the preferred instance fails.

Using Enterprise Manager, you can also specify an instance as Not Used. This setting means the service does not run on the instance, even if the preferred instance for the service fails.

See Also:

- "About FAN Callouts" on page 6-7
- "Creating Services" on page 6-17
- "About Workload Management with Dynamic Database Services" on page 6-1
- *Oracle Database High Availability Overview* for more information about the high availability products and features available for Oracle Database

About Service Failover in Policy-Managed Databases

When you specify that a service is UNIFORM, Oracle Clusterware attempts to ensure that the service always runs on all the available instances for the specified server pool. If the instance fails, then the service is no longer available on that instance. If the cardinality of the server pool increases and an instance is added to the database, then the service is started on the new instance. You cannot manually relocate the service to a specific instance.

When you specify that a service is SINGLETON, Oracle Clusterware attempts to ensure that the service always runs on only one of the available instances for the specified server pool. If the instance fails, then the service fails over to a different instance in the server pool. You cannot specify which instance in the server pool the service should run on.

For SINGLETON services, if a service fails over to a new instance, then the service is not moved back to its original instance when that instance becomes available again.

See Also:

- "About FAN Callouts" on page 6-7
- "Creating Services" on page 6-17
- "About Workload Management with Dynamic Database Services" on page 6-1
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about automatic workload management using dynamic database services

About Automatic Starting of Services

When you define a service, you can also define the management policy for that service. You can choose either an automatic or manual management policy:

- **Automatic**: The service always starts when the database starts.

  Note: When you use automatic services in an administrator-managed database, during planned database startup, services may start on the first instances that become available rather than their preferred instances.

- **Manual**: Requires you to start the service manually after the database starts.

See Also:

- "Creating Services" on page 6-17
- "About Workload Management with Dynamic Database Services" on page 6-1

About the Database Resource Manager

The Database Resource Manager is a database feature you can use to control the database resources allocated to users, applications, and services. This approach ensures that users, applications, and services receive their share of the available database resources. The Database Resource Manager enables an Oracle RAC database running on one or more nodes to support multiple applications and mixed workloads with optimal efficiency.

The Database Resource Manager provides the ability to prioritize work within an Oracle database or your Oracle RAC environment. For example, high priority users, such as online workers, would get more resources to minimize response time, while lower priority users, such as batch jobs or reports, would get fewer resources, and could take longer to run. Database Resource Manager enables more granular control over resources.

Resources are allocated to users according to a resource plan specified by the database administrator. The following terms are used in specifying a resource plan:

- A **resource plan** specifies how the resources are to be distributed among various users based on resource consumer groups.
- **Resource consumer groups** enable the administrator to group user sessions by resource requirements. Resource consumer groups are different from user roles; one database user can have different sessions assigned to different resource consumer groups.
- **Resource allocation methods** are the methods or policies used by the Database Resource Manager when allocating for a particular resource. Resource consumer groups and resource plans use the resource allocation methods. The database provides the resource allocation methods that are available, but the DBA determines which method to use.
- **Resource plan directives** are a means of assigning consumer groups to particular plans and partitioning resources among consumer groups by specifying parameters for each resource allocation method.
- **Subplans**, which the DBA can create within a resource plan, enable further subdivision of resources among different users of an application.
- **Levels** provide a mechanism to specify distribution of unused resources among available users. Up to eight levels of resource allocation can be specified.

The Database Resource Manager enables you to map a resource consumer group to a service so that users who connect using that service are members of the specified resource consumer group, and thus restricted to the resources available to that resource consumer group.

To learn more about managing the Database Resource Manager using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Administration** menu, select **Resource Manager**, and then **Getting Started**.

See Also:

- "About Workload Management with Dynamic Database Services" on page 6-1
- *Oracle Database Administrator's Guide* for more information about the Database Resource Manager

About Oracle RAC High Availability Framework

The Oracle RAC high availability framework enables Oracle RAC to maintain the database, components, and applications in a running state always. If an instance, component, or application fails, then it can be automatically restarted to keep Oracle Database operating at full capacity.

Oracle Database focuses on maintaining service availability. In Oracle RAC, Oracle services are designed to be continuously available with workloads shared across one or more instances. The Oracle RAC high availability framework maintains service availability by storing the configuration information for each service in the Oracle Cluster Registry (OCR). Oracle Clusterware recovers and balances services across instances based on the service definition.

See Also:

- "About Oracle Clusterware" on page 5-1
- "About Workload Management with Dynamic Database Services" on page 6-1
- *Oracle Database High Availability Overview* for information about determining the high availability requirements for your system

About Fast Application Notification (FAN)

One of the main requirements of a highly available application is for it to be quickly notified when something happens to critical system components. Notification enables the application to execute event-handling programs. The timely execution of such programs minimizes the time it takes to react to cluster resource organizations and the impact of cluster component failures by avoiding costly connection time outs and application time outs.

Fast Application Notification (FAN) is a high availability notification mechanism that Oracle RAC uses to notify other processes about cluster configuration and service-level information, including status changes such as `UP` or `DOWN` events. FAN `UP` and `DOWN` events can apply to instances, services, and nodes. FAN also publishes Load Balancing Advisory events.

FAN enables the automated recovery of applications when cluster components fail. For cluster configuration changes, the Oracle RAC high availability framework publishes a FAN event immediately after a change occurs regarding the state of the instances in the cluster. Instead of waiting for the application to query the database and detect a problem, applications can receive FAN events and react immediately.

FAN `UP` and `DOWN` events provide the following benefits:

- For `DOWN` events, the disruption to the application can be minimized because sessions that are connected to the failed instance or node can be terminated. Incomplete transactions can be terminated and the application user notified immediately. Application users who request connections are directed to instances that are started and are providing the requested service.

- For `UP` events, when services and instances are started, new connections can be created so that the application can immediately take advantage of the extra resources.

Oracle Clusterware and Oracle RAC use Oracle Notification Service (ONS) to propagate FAN messages both within the Oracle cluster and to client or mid-tier machines. ONS is installed with Oracle Clusterware and the resources to manage the ONS daemons are created automatically during the installation process. ONS daemons run on each node of the cluster and send and receive messages from a configured list of nodes where other ONS daemons are active; this list of nodes can include nodes outside the cluster, such as application server tiers or client nodes.

FAN can also be used with Oracle Data Guard, Active Data Guard, Oracle WebLogic Server Active GridLink for RAC, Universal Connection Pool (UCP) clients, Global Data Services, and OCI-based clients including OCI/OCCI, ODP.NET and OCI Session Pool, among others.

See Also:

- "About Connection Load Balancing"
- "About the Load Balancing Advisory"
- "About FAN Callouts"

About FAN Callouts

FAN callouts are server-side executable files that Oracle RAC runs immediately when high availability events occur. A callout is essentially a shell script or precompiled executable written in any programming language. Some examples of how you can use FAN callouts to automate the actions performed when events occur in a cluster configuration are as follows:

- Starting and stopping server-side applications
- Relocating low-priority services when high-priority services come online
- Sending text or numeric messages to pagers

The executable files for FAN callouts are stored in the *Grid_home*/racg/usrco subdirectory. If this subdirectory does not exist in your Grid home, then you must create this directory with the same permissions and ownership as the *Grid_home*/racg/tmp subdirectory.

All executables in the *Grid_home*/racg/usrco subdirectory are executed immediately, in an asynchronous fashion, when a FAN event is received through the Oracle Notification Service (ONS). For most event types, the callout is executed on one node in the cluster (the node generating the event), thus a copy of the executable files used by FAN callouts should be available on every node that runs Oracle Clusterware. An example of a callout script can be found in the section "Oracle RAC FAN Callout Start Services Script" in the following white paper, available on Oracle Technology Network: http://www.oracle.com/us/products/database/maa-wp-beehive-maa-130890.pdf

See Also:

- "About Connection Load Balancing" on page 6-13
- "About the Load Balancing Advisory" on page 6-12
- *Oracle Database Administrator's Guide* for more information about using FAN callouts
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring Fast Application Notification and FAN callouts

About Using Transaction Guard for Improving Client Failover

Transaction Guard provides a generic tool for applications to use for at-most-once execution in case of planned and unplanned outages and repeated submissions. Starting with Oracle Database 12c, applications can use a **logical transaction ID** (LTXID) to determine the outcome of the last transaction open in a database session following an outage. Without using Transaction Guard, applications that attempt to retry operations following outages can cause logical corruption by committing duplicate transactions.

Without Transaction Guard, one of the fundamental problems for recovering applications after an outage is that the commit message that is sent back to the client is not durable. If there is a break between the client and the server, the client sees an error message indicating that the communication failed (also called a **recoverable error**). This error does not inform the application whether the submission executed any commit operations, or if a procedural call ran to completion while executing all expected commits. The error also does not indicate session state changes or intermittent failures. Worse yet, the application may still be running disconnected from the client.

These recoverable errors can cause end users or applications that have their own replay (instead of using Application Continuity) to issue duplicate transaction submissions, re-issue changes that are already committed in the database, and other forms of "logical corruption." The transaction cannot be validly resubmitted if the non-transactional state is incorrect or if it is committed. Continuing to process a committed but not completed call can result in the application using a database session that is in the wrong state.

In Oracle Database 12*c* Release 1 (12.1), Transaction Guard provides new, integrated tools for applications to use to achieve idempotence automatically and transparently, and in a manner that scales. Transaction Guard uses the **logical transaction ID** (LTXID) to eliminate duplicate transactions, a process referred to as **transaction idempotence**. The logical transaction id persists on commit and is reused following a rollback. During normal runtime, a LTXID is automatically held in the session at both the client and server for each database transaction. At commit, the LTXID persists as part of committing the transaction and the database returns the next LTXID to use to the client.

An example of how Application Continuity or an application uses Transaction Guard:

1. The Oracle Client driver receives a FAN down event.
2. FAN automatically aborts the terminated session.
3. If it is a **recoverable error**, then:

a. Get the last LTXID from the failed session using the APIs provided by the client driver: `getLogicalTransactionId()` for JDBC, `LogicalTransactionId` for ODP.NET, and `OCI_ATTR_GET` for OCI.

b. Obtain a new session.

c. Call the PL/SQL procedure `DBMS_APP_CONT.GET_LTXID_OUTCOME` with the last LTXID gotten from the failed session.

Note: You must grant the `EXECUTE` privilege on the `DBMS_APP_CONT` package to the database users that use the package.

d. If the outcome of this transaction is COMMITTED and COMPLETED, then return the result of that transaction to the application.

e. If the outcome of the session state (such as messages or the PL/SQL block) is COMMITTED, but not COMPLETED, then return the result of that transaction to the application and a warning that the transaction was committed, but there may be additional states such as row counts preventing the application from continuing.

Note: Most applications will not need these states. This is a consideration for those that do. For those that do not care about row counts or other states from the call, then return COMMITTED to the application.

f. If the outcome of the transaction is not COMMITTED, then, the user or application can chose to resubmit the last request.

See Also:

- *Oracle Database Development Guide* for more information about Transaction Guard and idempotence
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about Transaction Guard and client failover for Oracle RAC
- *Oracle Database JDBC Developer's Guide* about using Transaction Guard with Java applications
- *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows* for information about Transaction Guard support for ODP.NET applications

About Application Continuity for Masking Outages

Following planned and unplanned outages, **Application Continuity** attempts to mask the outage by rebuilding the database session, and resubmitting the pending work following recoverable errors that make the database session unavailable. When Application Continuity is configured, an end-user request is replayed if it did not complete (if replay is enabled when the outage occurred); replay is started if the time has not exceeded the replay timeout attribute specified for the service. When a component fails or becomes unresponsive, Application Continuity attempts to restore the database sessions back to the current time. When replay succeeds, this feature masks applications from transient outages (such as session failure, instance or node outages, network failures and so on) and from planned outages such as repairs, configuration changes and patching the software.

Preparing to Use Application Continuity

Before you can use Application Continuity, you must configure attributes for the **database services** used by the applications. The application may also need to be modified to borrow and return connections rather than pin connections. This is normally a property. Request boundaries need to be added if one of the integrated pools (UCP, WLS data source) is not being used. You may also need to mark request boundaries if not returning connections to the Oracle Connection Pool and a property is not available to unpin. With Oracle Database 12*c* Release 1 (12.1), Application Continuity for Java is available for general use with the JDBC-Thin Oracle driver, the JDBC Universal Connection Pool, and WebLogic Server Active Grid Link. It is also available with OCI 12*c* when using PeopleSoft Applications.

To use Application Continuity with an Oracle RAC database, use the following configuration checklist:

❑ Service Configuration Checks:

  – Create a dynamic database service (also referred to as an *application service*) to connect to the database. Do not use the Oracle SID, instance name, or the default database service to connect to an Oracle RAC or Oracle Data Guard database.

  – For the service, set `failovertype` to `TRANSACTION`, `commit_outcome` to `TRUE` and `notification` to `TRUE`. Optionally, to find the best connections to use, set `rlb_goa` to `SERVICE_TIME` and `clb_gloal` to `SHORT`. See "Creating Services Using SRVCTL" on page 6-20.

❑ Software Configuration Checks (Database and Middle-tier):

  – Use Oracle Database 12*c* Release 1 (12.1) or later.

  – Use the JDBC Universal Connection Pool (12.1) or WebLogic Active GridLink (12.1.2 or later) configured with the JDBC Replay data source. You can also use the JDBC Replay data source with your own JDBC connection pool.

  – If a statement cache at the application server level is enabled (for example, the WebLogic or third-party application server statement cache), then this *must* be disabled when the replay is used. Instead, configure the JDBC statement cache, which supports Application Continuity. The JDBC statement cache also performs better because it is optimized for JDBC and Oracle. Use `oracle.jdbc.implicitstatementcachesize=`*nnn*.

  – Use FAN and FCF for WebLogic Active GridLink Data Source, Universal Connection Pool, or, as an option, use simple JDBC FAN with third party pool.

  – Check the resource requirements; ensure there is sufficient CPU and memory at the middle tier.

  Application Continuity is managed on the server-side, and uses hardware when available for CRCs. On the client side, additional CPU costs are incurred for garbage collection. The replay driver requires more memory than the base driver because the calls are retained until the end of a request. At the end of the request, the calls are released to the garbage collector. This action differs from the base driver that releases closed calls.

> **Note:** CPU overhead is reduced on the database side for platforms with current Intel and Sparc chips where validation is assisted by the hardware.

  - Determine whether the application can use **mutables**, or sequential data that might be different on a different instance. If so, then you configure the application to keep the original values for `SYSDATE`, `SYSTIMESTAMP`, `SYS_GUID`, and sequences during failover.

    See *Oracle Database SQL Language Reference* for information on issuing the `GRANT` privilege to users for `KEEP [DATE TIME|SYSGUID|`*`Sequence`*`]`.

  - Grant `EXECUTE` on the DBMS_APP_CONT PL/SQL package to the users that will use Application Continuity.

  - After consulting the application certification, assign the `GRANT` privilege for keeping mutables to the users that will use replay.

  - Check with the application developers to determine if any requests should not be replayed. The application must explicitly call an API to disable replay for those requests.

- ❏ Application Code Checks (consult with the application developer):

  - Determine whether the application uses Oracle JDBC concrete classes. If it does, and if access is needed to Oracle proprietary APIs, then plan to replace these concrete classes with standard JDBC or Oracle JDBC interfaces. Refer to My Oracle Support Note 1364193.1 at :

    `https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1364193.1`

  - Use a connection string that sets connection retries and a delay between these retries. For an example of configuring the connection properties using JDBC, see "Configuring JDBC Clients for Fast Connection Failover".

  - Decide whether an optional callback will be used for initializing connections from the application to the database. When using Oracle WebLogic Server or the Universal Connection Pool, connection labeling is recommended. If registered, a callback will execute at runtime and replay.

  - Determine whether replay must be explicitly disabled for any code path in the application. In particular, review external PL/SQL calls such as messaging and callouts and server side Java. If these should not be replayed, then use the `disableReplay` API.

  - Determine whether the application must have begin and end request API's added to the application's own connection pool to identify request boundaries or if the application borrows and returns connections from the WebLogic Server Pool or Universal Connection Pool for each request.

    If the application borrows and returns connections from the WebLogic Server Pool or Universal Connection Pool for each request, then no change is needed.

    If the application uses Oracle connection pools, and does not return connections between requests, see if there is a property to set to return connections to the pool. If there is no property to set, or if using the application's own connection pool, then you must add `beginRequest` and `endRequest` boundaries.

See Also:

- *Oracle Database Development Guide* for more information on how Application Continuity works, and how you can use it in applications
- *Oracle Database Concepts* for more information about transactions
- *Oracle Database JDBC Developer's Guide* for more information about Application Continuity for Java applications
- "Creating Services" on page 6-17

Using Application Continuity for Planned Outages

Application Continuity is recommended for planned outages combined with draining requests from Oracle connection pools. Using any FAN-aware pool with FAN configured (OCI, UCP, DRCP, WebLogic Server, or ODP.NET), the steps are as follows:

1. Use the SRVCTL `relocate` command to relocate the service from the instance being shut down without disrupting the sessions (do not use the `-force` parameter). If relocating a uniform service, then use `srvctl stop service` at the instance (do not use the `-force` parameter).
2. The FAN event for the planned outage clears the idle sessions immediately and marks the active sessions to be released at **check in** (end of request). This drains the sessions from the instance without disrupting work.
3. All sessions should check in, but in reality this does not always happen. When sufficient time has elapsed for the sessions to check in, stop the instance using the `ABORT` shutdown option. For Application Continuity enabled pools (UCP and WebLogic Server), Application Continuity attempts to recover those remaining sessions that did not check in.

   This step applies for UCP and WLS, and any pool that adds `beginRequest/endRequest` on JDBC Thin Replay Driver.
4. Restart the instance and service. Run-time load balancing, if enabled, balances the sessions back to use the restored instance at the next request boundaries.

About the Load Balancing Advisory

The Load Balancing Advisory provides information to applications or clients about the current service levels that the Oracle RAC database instances are providing. Applications can take advantage of the load balancing Fast Application Notification (FAN) events to direct work requests to the instance in the cluster that provides the best performance based on the workload management directives that you have defined for that service. Also, when an instance is restarted, Oracle RAC uses FAN events to notify the application's connection pool so that the connection pool can create connections to the recently started instance and take advantage of the additional resources that this instance provides.

The load balancing advisory is integrated with the Automatic Workload Repository built into Oracle Database 12*c*. The Automatic Workload Repository measures response time and CPU consumption for each service.

The advice given by the Load Balancing Advisory takes into account the power of the server and the current workload of the service on the server. Enabling the Load Balancing Advisory helps improve the throughput of applications by not sending work to instances that are overworked, running slowly, not responding, or have failed.

Your application can take advantage of the Load Balancing Advisory without any programmatic changes if you use an integrated Oracle client, one that has the Run-time Connection Load Balancing feature. Due to the integration with FAN, Oracle integrated clients are more aware of the current status of an Oracle cluster. This prevents client connections from waiting or trying to connect to an instance that is no longer available. The integrated clients for FAN events include the Universal Connection Pool (UCP) used by Oracle Database 12*c* JDBC, Oracle Database 12*c* ODP.NET, and Oracle Database 12*c* Oracle Call Interface (OCI).

You configure your Oracle RAC environment to use the Load Balancing Advisory by defining service-level goals for each service used. Defining a service-level goal enables the Load Balancing Advisory for that service and enables the publication of FAN load balancing events. There are two types of service-level goals for Run-time Connection Load Balancing:

- Service Time. The Load Balancing Advisory attempts to direct work requests to instances according to their response time. Load Balancing Advisory data is based on the elapsed time for work done by connections using the service, and the available bandwidth to the service. This goal is best suited for workloads that require varying lengths of time to complete, for example, an internet shopping system.
- Throughput. The Load Balancing Advisory measures the percentage of the total response time that the CPU consumes for the service. This measures the efficiency of an instance, rather than the response time. This goal is best suited for workloads where each work request completes in a similar amount of time, for example, a trading system.

If you do not select the Enable Load Balancing Advisory option, then the service-level goal is set to None, which disables load balancing for that service.

See Also:

- "About Fast Application Notification (FAN)" on page 6-6
- "Configuring Clients for High Availability" on page 6-23
- "Administering Services" on page 6-21
- "About Workload Management with Dynamic Database Services" on page 6-1
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about integrated Oracle clients

About Connection Load Balancing

Oracle Net is a software component that resides on the client and on the Oracle database server. It establishes and maintains the connection between the client application and the server, and exchanges messages between them using industry standard protocols. For the client application and a database to communicate, the client application must specify location details for the database it wants to connect to, and the database must provide some sort of identification or address.

On the database server, the Oracle Net Listener, commonly known as the listener, is a process that listens for client connection requests. The configuration file for the listener is the `listener.ora`.

Oracle Database 12*c* database clients use **SCAN** and the easy connect method to connect to the database. SCAN can resolve to multiple IP addresses, reflecting multiple listeners in the cluster handling public client connections. When using the easy connect method, you do not have to configure any client network files. You simply specify a connect identifier with the following format:

```
SCAN[:port]/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.

You can also use Net Configuration Assistant (NETCA) to create a **net service name**. The net service name resolves to the **connect descriptor**. The address portion of the connect descriptor is actually the protocol address of the **listener**. The client uses a connect descriptor to specify the database or instance to which the client wants to connect.

When a net service name is used, establishing a connection to a database instance takes place by first mapping the net service name to the connect descriptor. This mapped information is stored in one or more repositories of information that are accessed using naming methods. The most commonly used naming method is Local Naming, where the net service names and their connect descriptors are stored in a localized configuration file named tnsnames.ora.

When the client connects to the cluster database using a service, you can use the Oracle Net connection load balancing feature to spread user connections across all the instances that are supporting that service. There are two types of load balancing that you can implement: client-side and server-side load balancing. In an Oracle RAC database, client connections should use both types of connection load balancing. When you create an Oracle RAC database using Oracle Database Configuration Assistant (DBCA), DBCA configures and enables server-side load balancing by default.

See Also:

- "About Workload Management with Dynamic Database Services"
- "Verifying Oracle Net Supports Newly Created Services" on page 6-22
- *Oracle Database 2 Day DBA* for more information about network configurations
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*

About Client-Side Load Balancing

Client-side load balancing balances the connection requests across the listeners. When the listener receives the connection request, the listener connects the user to an instance that the listener knows provides the requested service.

Client-side load balancing is defined in your client connection definition by setting the parameter LOAD_BALANCE=yes in the tnsnames.ora file. When you set this parameter to yes, the Oracle client randomly selects an address from the address list, and connects to that node's listener. This balances client connections across the available listeners in the cluster.

When you create an Oracle RAC database using DBCA, the assistant creates a sample load balancing connection definition in the tnsnames.ora file.

Client-side load balancing includes connection failover. With connection failover, if an error is returned from the chosen address, then Oracle Net Services tries the next address in the address list until either a successful connection is made or it has exhausted all the addresses in the list.

See Also:

- "About Connection Load Balancing" on page 6-13
- "About the Load Balancing Advisory" on page 6-12
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*

About Server-Side Load Balancing

With server-side load balancing, the listener directs a connection request to the best instance currently providing the service by using information from the Load Balancing Advisory.

For each service, you can define the method the listener uses for load balancing by setting the connection load balancing goal. You can use a goal of either long or short for connection load balancing. These goals have the following characteristics:

- Short—Connections are distributed across instances based on the best response time. Use the Short connection load balancing goal for applications that have connections of brief duration.
- Long—Connections are distributed across instances based on the number of sessions in each instance, for each instance that supports the service. Use the Long connection load balancing goal for applications that have connections of long duration. This is typical for connection pools and SQL*Forms sessions. Long is the default connection load balancing goal.

Note: If you did not use DBCA to create your database, or if you are using listener ports other than the default of 1521, then you must configure the `LOCAL_LISTENER` and `REMOTE_LISTENER` database initialization parameters for your cluster database to point to *`SCAN:port`*.

See Also:

- "About Connection Load Balancing" on page 6-13
- "About the Load Balancing Advisory" on page 6-12

About Run-time Connection Load Balancing

Run-time Connection Load Balancing is a feature of Oracle connection pools that can distribute client work requests across the instances in an Oracle RAC database based on the Load Balancing Advisory information. The connection allocation is based on the current performance level provided by the database instances as indicated by the Load Balancing Advisory FAN events. This provides load balancing at the transaction level, instead of load balancing at the time of the initial database connection.

With Run-time Connection Load Balancing, applications use Load Balancing Advisory information to provide better performance to users. OCI Session pools and ODP.NET connection pools support Run-time Connection Load Balancing. For Java applications, Oracle recommends the Universal Connection Pool (UCP). The Universal Connection

Pool is integrated to take advantage of Load Balancing Advisory information. UCP, introduced in Oracle Database 11*g* patch set 1 (11.1.0.7), can be used against Oracle Database 10*g*, Oracle Database 11*g*, or Oracle Database 12*c*.

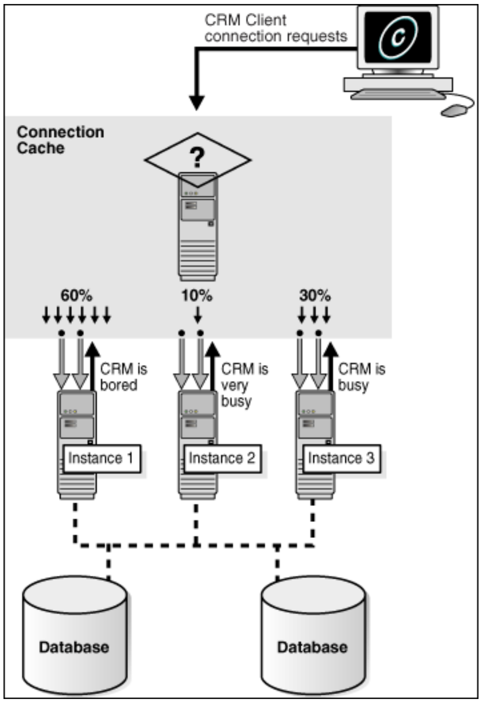
You must enable the client data source for Run-time Connection Load Balancing with a service that has the following configuration:

- The Load Balancing Advisory is enabled and the service-level goal is set to either Service Time or Throughput.
- The service connection load balancing goal is set to Short.

The illustration in the steps below illustrates Run-time Connection Load Balancing. In this illustration, the Oracle RAC database has three instances. Suppose that the Load Balancing Advisory indicates that Instance1 and Instance3 have the best performance, while Instance2 currently has less than optimal performance. When Run-time Connection Load Balancing is enabled for the Universal Connection Pool, the following process occurs:

1. A client requests a connection from the connection pool.
2. Run-time Connection Load Balancing selects the connection that belongs to the most efficient (best) instance from the connection pool. In the illustration in the next step, there are three possible nodes to which the connection can be routed. Instance1, which has the least amount of CPU workload, is currently being assigned about 60 percent of the incoming connections. Instance2, which is currently overloaded, is only being assigned around 10 percent of the incoming connections. Instance3, which has a high workload, is being assigned around 30 percent of the incoming connections. The best instance to handle the connection request in this case would be Instance1.
3. The client receives the connection that would process the work request with the best response time.

Figure 6–1 Run-time Connection Load Balancing



Starting with Oracle Database 11*g* you can use an additional flag in the load balancing advisory event called **affinity hint**. The affinity hint is a flag that indicates if affinity is a benefit for a particular instance and service combination. Different instances offering the same service can have different settings for the affinity hint.

The affinity hint is automatic when the load balancing advisory is turned on through setting the goal on the service. This flag is for temporary affinity that lasts for the duration of a web session. Web conversations often connect and disconnect many times during the entire session. During each of these connects, it may access the same or similar data, for example, a shopping cart, Siebel, and so on. Affinity can improve buffer cache efficiency, which lowers CPU usage and transaction latency.

Applications using Oracle Database 12*c* and UCP can take advantage of this new affinity feature. If the affinity flag is turned on in the Load Balancing Advisory event, then UCP creates an Affinity Context for the web session such that when that session does a get connection from the pool, the pool always tries to give it a connection to the instance it connected to the first time it acquired a session. The choice of instance for the first connection is based on the current load balancing advisory information.

See Also:

- "About Connection Load Balancing" on page 6-13
- "About the Load Balancing Advisory" on page 6-12
- "About FAN Callouts" on page 6-7

Creating Services

To manage workloads, you can define **database services** that you assign to a particular application or to a subset of an application's operations. You can also use services to manage the workload for different types of work. You can create a service using Oracle Enterprise Manager or the SRVCTL utility.

- Creating Services Using Enterprise Manager
- Creating Services Using SRVCTL

Creating Services Using Enterprise Manager

You can use either Oracle Enterprise Manager Cloud Control or Oracle Enterprise Manager Database Express to create services.

To create a service using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Cluster Managed Database Services**. Enter or confirm the credentials for the Oracle RAC database and host operating system and click **Continue**.

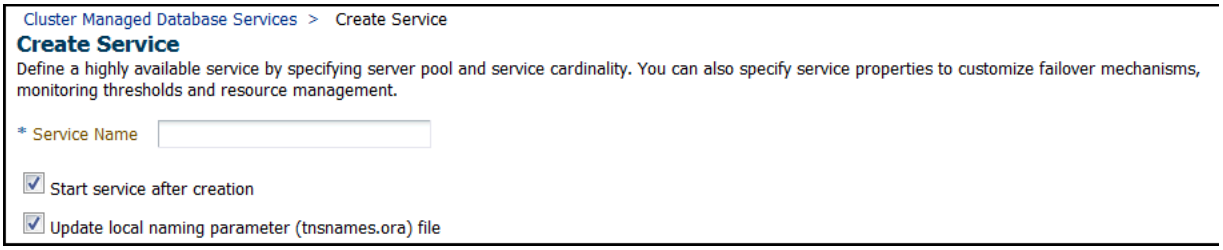
   The Cluster Managed Database Services page appears.

3. Click **Create Service**.

   The Create Service page appears.

4. Enter the name of your service in the Service Name field, for example, `DEVUSERS`.

5. Select **Start Service after creation** if you want the service to be started after it is created. Select **Update local naming parameter (tnsnames.ora) file** if you want Oracle Enterprise Manager Cloud Control to add the new service to the local Oracle Net Services `tnsnames.ora` file.



Cluster Managed Database Services > Create Service

Create Service

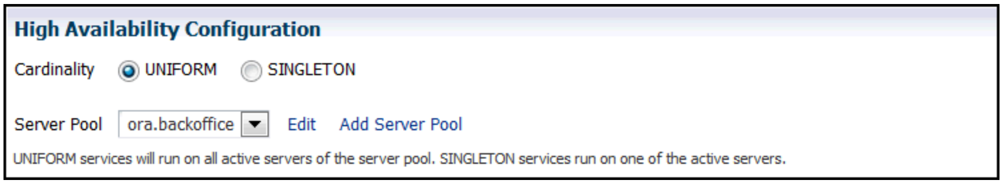
Define a highly available service by specifying server pool and service cardinality. You can also specify service properties to customize failover mechanisms, monitoring thresholds and resource management.

* Service Name

☑ Start service after creation

☑ Update local naming parameter (tnsnames.ora) file

6. (Policy-managed databases only) For the service type, select **UNIFORM** or SINGLETON.



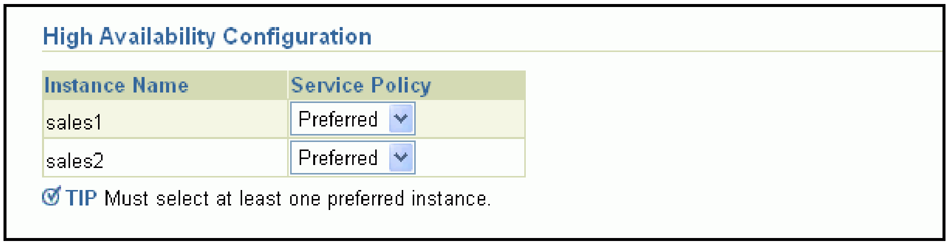
High Availability Configuration

Cardinality ● UNIFORM ○ SINGLETON

Server Pool ora.backoffice

UNIFORM services will run on all active servers of the server pool. SINGLETON services run on one of the active servers.

7. (Administrator-managed databases only) For each instance choose whether that instance is a Preferred instance or an Available instance. If you do not want the service to run on an instance under any circumstances, then set the Service Policy for that instance to Not Used.



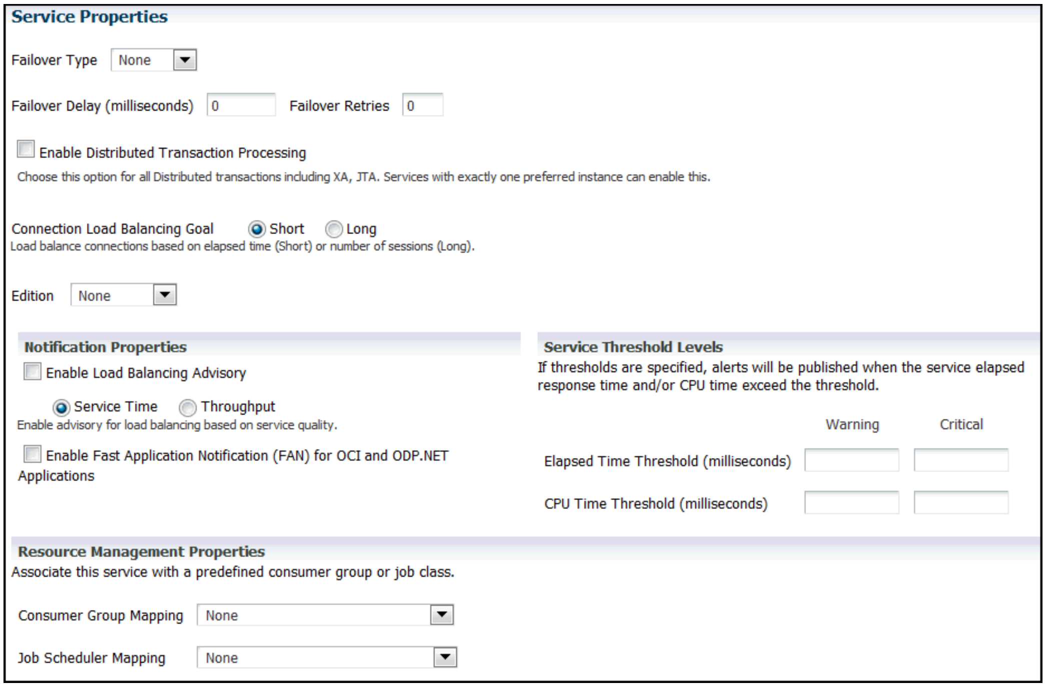
High Availability Configuration

Instance Name	Service Policy
sales1	Preferred
sales2	Preferred

TIP Must select at least one preferred instance.

8. In the Service Properties section, select **Short** for Connection Load Balancing Goal to distribute the connection workload based on elapsed time instead of the overall number of connections. Otherwise, choose **Long**.

9. Select **Enable Load Balancing Advisory** under the sub-heading Notification Properties to enable the Load Balancing Advisory for this service, as shown in the following screenshot. Choose a service-level goal of either **Service Time** or **Throughput**.



Service Properties

Failover Type: None

Failover Delay (milliseconds): 0   Failover Retries: 0

□ Enable Distributed Transaction Processing

Choose this option for all Distributed transactions including XA, JTA. Services with exactly one preferred instance can enable this.

Connection Load Balancing Goal ● Short ○ Long

Load balance connections based on elapsed time (Short) or number of sessions (Long).

Edition: None

Notification Properties

□ Enable Load Balancing Advisory

● Service Time ○ Throughput

Enable advisory for load balancing based on service quality.

□ Enable Fast Application Notification (FAN) for OCI and ODP.NET Applications

Service Threshold Levels

If thresholds are specified, alerts will be published when the service elapsed response time and/or CPU time exceed the threshold.

	Warning	Critical
Elapsed Time Threshold (milliseconds)		
CPU Time Threshold (milliseconds)		

Resource Management Properties

Associate this service with a predefined consumer group or job class.

Consumer Group Mapping: None

Job Scheduler Mapping: None

10. Select **Enable Fast Application Notification** under the heading Notification Properties if an Oracle Call Interface (OCI) or ODP.NET application uses this service, and you want to enable FAN.

11. In the Service Threshold Levels section, you can optionally set the service-level thresholds by entering a value in milliseconds for Warning and Critical thresholds for the Elapsed Time and CPU Time metrics.

12. If you want to use a Resource Plan to control the resources used by this service, then select the name of the consumer group from the Consumer Group Mapping list in the Resource Management Properties section. For example, you might choose the `LOW_GROUP` consumer group to give development users low priority to database resources.

    Note: You cannot change the consumer group name for a service on the Edit Service page. This is because there may be several consumer groups associated with a given service. However, the Edit Service page contains a link to the Resource Consumer Group Mapping page, where you can modify the consumer group mapping for the service.

13. If a specific Oracle Scheduler job class uses this service, then you can specify the mapping by selecting the name from the Job Scheduler Mapping list in the Resource Management Properties.

14. Click **OK** to create the service.

See Also:

- "About Workload Management with Dynamic Database Services" on page 6-1
- "About Connection Load Balancing" on page 6-13
- "About the Load Balancing Advisory" on page 6-12
- "About Fast Application Notification (FAN)" on page 6-6
- "Administering Services" on page 6-21
- *Oracle Database Administrator's Guide* for more information about the Oracle Scheduler

Creating Services Using SRVCTL

SRVCTL is a command-line interface that you can use to create services and administer Oracle Database objects that Oracle Clusterware manages. For functionality that is not available with Oracle Enterprise Manager Database Express, you can use SRVCTL commands instead. See *Oracle Real Application Clusters Administration and Deployment Guide* for information about how to use the SRVCTL utility.

To configure services for Application Continuity or Transaction Guard using SRVCTL:

1. To configure services for Application Continuity, you must set the service attributes `failovertype` to `TRANSACTION`, `commit_outcome` to `TRUE`, and `notification` to `TRUE`. Additionally, you can set values for these other service attributes for Application Continuity and load balancing:

   - `replay_init_time`: Specifies the time in seconds after which replay will not be started; example settings are 60, 300, or 600 seconds; the recommended value is 900 seconds.
   - `retention`: Specifies the time (in seconds) that the commit outcome is retained for Transaction Guard. Recommended value is 86400 (24 hours).
   - `failoverretry`: Number of connection retries for each replay attempt; recommended value is 30.
   - `failoverdelay`: Delay in seconds between connection retries; recommended value is 10.
   - `clbgoal`: For connection load balancing, use `SHORT` when using run-time load balancing.
   - `rlbgoal`: Set to `SERVICE_TIME`.

   To use SRVCTL to modify the service attributes, use a command similar to the following, where *`racdb`* is the name of your Oracle RAC database, and *`app1`* is the name of the service you are modifying:

   ```
   srvctl modify service -db racdb -service app1 -clbgoal SHORT
   -rlbgoal SERVICE_TIME -failoverretry 30 -failoverdelay 10
   -failovertype TRANSACTION -replay_init_time 1800 -retention 86400
   -notification TRUE commit_outcome TRUE
   ```

2. To create a service for Application Continuity for a policy-managed Oracle RAC database, use a command similar to the following, where *`racdb`* is the name of your Oracle RAC database, *`app2`* is the name of the service you are modifying, and *`Svrpool1`* is the name of the server pool in which the service is offered:

```
srvctl modify service -db racdb -service app2 -serverpool ora.Srvpool1
-clbgoal SHORT -rlbgoal SERVICE_TIME -failoverretry 30
-failoverdelay 10 -failovertype TRANSACTION -replay_init_time 1800
-retention 86400 -notification TRUE -commit_outcome TRUE
```

3. To enable Transaction Guard, and but not Application Continuity, create the service with `commit_outcome` = TRUE. You can also use the attribute `retention` to specify how long in seconds to maintain the history. Additionally, it is recommended that you set the attribute `notification` to TRUE, for FAN events.

   To use SRVCTL to modify the service attributes, use a command similar to the following, where *racdb* is the name of your Oracle RAC database, and *app2* is the name of the service you are creating:

   ```
   srvctl modify service -db racdb -service app2 -commit_outcome TRUE -retention
   86400 -notification TRUE
   ```

Administering Services

You can create and administer services using Enterprise Manager. You can also use the SRVCTL utility to perform most service management tasks.

- About Service Administration Using Enterprise Manager
- Using the Cluster Managed Database Services Page
- Verifying Oracle Net Supports Newly Created Services

See Also:

- "About Oracle Services" on page 6-2
- "Creating Services" on page 6-17

About Service Administration Using Enterprise Manager

The Cluster Managed Database Services page is the master page for beginning all tasks related to services. To access this page, go to the Cluster Database Maintenance page, then click **Cluster Managed Database Services** in the Services section. You can use this page and links from this page to do the following:

- View a list of services for the cluster.
- View the instances on which each service is currently running.
- View the status for each service.
- Create or edit a service.
- Start or stop a service.
- Enable or disable a service.
- Perform instance-level tasks for a service.
- Delete a service.

See Also:

- "Administering Services" on page 6-21
- "About Oracle Services" on page 6-2
- "Creating Services" on page 6-17

Using the Cluster Managed Database Services Page

When managing services using Enterprise Manager, you use the Cluster Managed Database Services page.

On the Cluster Managed Database Services page you can perform the following tasks:

- View a list of services for the cluster, the instances on which each service is currently running, the status of each service, the server pool associated with the service, the responsiveness of the service, and any alerts or messages for that service.
- Start or stop a service.
- Test the connection for a service, or display the Oracle Net TNS strings being used.
- Access the Create Service page.
- Edit, delete, enable, or disable a service.
- Access the Edit Server Pool page for the server pool associate with each service.

To access the Cluster Managed Database Services page:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Cluster Managed Database Services**. Enter or confirm the credentials for the Oracle RAC database and host operating system and click **Continue**.

   The Cluster Managed Database Services page appears and displays the services that are available on the cluster database instances.

> **See Also:**
>
> - "About Service Administration Using Enterprise Manager" on page 6-21
> - "About Oracle Services" on page 6-2
> - "Creating Services" on page 6-17

Verifying Oracle Net Supports Newly Created Services

By configuring and using a SCAN for your cluster, you no longer have to modify the network setup on each client. This section shows you how to verify that the new service is recognized by Oracle Net and available to the database clients.

> **Note:** When using utilities to manage your cluster, databases, database instances, Oracle Automatic Storage Management (Oracle ASM), and listeners, use the appropriate binary that is in the home directory of the object or component you are managing. Also, set your `ORACLE_HOME` environment variable to point to this directory.
>
> For example, to use `lsnrctl` to manage the SCAN listener, use the binaries located in the Grid home (where the listener is running), and set the `ORACLE_HOME` environment variable to the location of the Grid home.

To verify Oracle Net Services supports the newly created service:

1. Determine if the listener on the local node recognizes the new service by using the following command:

   ```
   lsnrctl status LISTENER_SCAN1
   ```

   You should see a list for the new service, similar to the following:

   ```
   Services Summary...
   Service "DEVUSERS.example.com" has 2 instance(s).
     Instance "sales1", status READY, has 1 handler(s) for this service...
     Instance "sales2", status READY, has 1 handler(s) for this service...
   ```

   The displayed name for your newly created service, for example `DEVUSERS.example.com`, is the service name you use in your connection strings.

2. Test the Oracle Net Services configuration by attempting to connect to the Oracle RAC database using SQL*Plus and the SCAN. The connect identifier for easy connect naming has the following format:

   ```
   "[//]SCAN[:port]/service_name"
   ```

   SCAN is the SCAN for your cluster, which defaults to *cluster_name*.*GNS_sub_domain*. The *service_name* is the name of the database service you want to use for connecting to the database. You can optionally specify the TCP port number on which the Oracle SCAN listener listens for connections.

   For example, you might use the following commands to connect to the DEVUSERS service for Oracle RAC database in the `docrac` cluster, which belongs to the `example.com` domain:

   ```
   $ sqlplus system@\"docrac.example.com/DEVUSERS.example.com\"
   Enter password: password
   ```

   After you enter the password, you should see a message indicating you are successfully connected to the Oracle RAC database. If you get an error message, then examine the connect identifier and verify the user name, password, and service name were typed in correctly and all the information is correct for your Oracle RAC environment.

   See Also:

   - "About Connection Load Balancing" on page 6-13
   - *Oracle Database 2 Day DBA*
   - *Oracle Database 2 Day DBA*

Configuring Clients for High Availability

There are three central elements to consider when automating failover for application clients:

- First, clients that are connected at the time of failure must be quickly and automatically notified that a failure has occurred to avoid waiting for TCP/IP network time-outs before attempting to connect to the new database instance (such time-outs range anywhere from eight minutes to two hours, depending on operating system). Oracle RAC configurations use Fast Application Notification (FAN) to notify JDBC clients, OCI clients, and ODP.NET clients. FAN event notifications and callouts enable automatic and fast redirection of clients after a site failure.

- The second central element of client failover, is the redirection of new incoming requests to the new instance after a failure has occurred, which can be implemented using application services. When you create application services in an Oracle RAC database, if an instance to which you have assigned a service becomes unavailable, then the service is already available at another Oracle RAC instance or Oracle RAC relocates the service to an available instance of the database. Applications need only provide the name of the dynamic database service in their connect strings–there is no need to specify an instance name because, using listener cross registration, SCAN listeners and other remote listeners in the cluster are aware of which instances are currently providing a service when a connection request comes in.
- The third central element is masking the outage from the client and the application. Masking outages of the database session is a complex task for application development and, therefore, errors and timeouts are often exposed to the client. Application Continuity attempts to mask outages from applications by replaying incomplete application requests after unplanned and planned outages.

For applications that are unable to take advantage of Application Continuity, two additional features are available: Transparent Application Failover (TAF) for OCI based applications and Transaction Guard. Transaction Guard enables applications to build their own masking of outages by knowing the outcome of the last in-flight transaction.

This section deals with configuring high availability for application clients, and contains the following topics:

- Configuring Oracle Net Services Parameters For Highly Available Connections
- Configuring JDBC Clients for High Availability
- Configuring OCI Clients for High Availability
- Configuring ODP.NET Clients for High Availability

See Also:

- "About Fast Application Notification (FAN)" on page 6-6
- "About Oracle Services" on page 6-2
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about client-side connection configuration

Configuring Oracle Net Services Parameters For Highly Available Connections

For applications that need to connect to an Oracle Database, you can use the following Oracle Net Services parameters to control the amount of time to wait when attempting to connect to an address, the amount of time allowed for a client to establish a TCP connection to the database, and the number of times the application attempts to connect to the database:

- `CONNECT_TIMEOUT`
- `RETRY_COUNT`
- `TRANSPORT_CONNECT_TIME`

When using SCAN listeners in an Oracle RAC environment, with `FAILOVER = on`, setting the `RETRY_COUNT` parameter to 2 means the three SCAN IP addresses are traversed three times each, such that there are nine connect attempts (3 * 3).

You can specify these parameters in either the timeout section of the tnsnames.ora file or at the DESCRIPTION level of a connect string, for example:

```
net_service_name=
 (DESCRIPTION=
  (CONNECT_TIMEOUT=60)(RETRY_COUNT=3)(TRANSPORT_CONNECT_TIMEOUT=3
  (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=docrac.example.com)(PORT=1521))
   (CONNECT_DATA=
   (SERVICE_NAME=racdb.example.com)))
```

Note:

- The CONNECT_TIMEOUT parameter is equivalent to the sqlnet.ora parameter sqlnet.OUTBOUND_CONNECT_TIMEOUT and overrides it.
- You should not configure the sqlnet.outbound_connection_timeout parameter on the server sqlnet.ora file, as this can impact Oracle Clusterware operations.
- The TRANSPORT_CONNECT_TIMEOUT parameter is equivalent to the sqlnet.ora parameter tcp.CONNECT_TIMEOUT and overrides it.

See Also: *Oracle Database Net Services Reference* for more information about the timeout and retry parameters for Oracle Net Services

Configuring JDBC Clients for High Availability

An application that uses JDBC connections should use a connection pool, so an existing connection can be used in the pool instead of making a new connection to the database every time you access data. The Universal Connection Pool (UCP) is a Java-based connection pool that supports any type of connection (JDBC, LDAP, JCA), to any type of database (Oracle or non-Oracle) with any middle tier (Oracle or non-Oracle). It also supports standalone deployments such as TopLink or BPEL. UCP also supports integration features of Oracle Database such as Fast Connection Failover, Run-time Connection Load Balancing, Web Affinity, Application Continuity, and Transaction Guard with Oracle RAC when using the JDBC Thin driver.

Without Fast Connection Failover, if a node or network goes down and the application attempts to connect to that instance, then the connection request might hang for several minutes waiting for the TCP time out to be reached. If you chose not to use a connection pool with the integrated FCF feature, then your application can still be notified of Oracle RAC high availability events by configuring the Connection Failure Notification feature.

Application Continuity provides a general purpose, application-independent infrastructure that enables recovery of work from an application perspective, after the occurrence of a planned or unplanned outage. The outage can be related to the system, communication, or hardware layer following a repair, a configuration change, or a patch application. Application Continuity restores the database session by replaying the non-transactional and transactional user calls to reconstruct the database session as if the outage had not occurred. Application Continuity performs this replay beneath the application so that the outage appears to the application as a delayed execution.

Transaction Guard is a reliable protocol and tool that returns the outcome of the last in-flight transaction after an outage makes the database session unavailable.

Transaction Guard avoids the costs of ambiguous errors that lead to end user frustration, customer support calls, and lost opportunities.

This section explains how to perform the following tasks:

- Configuring JDBC Clients for Fast Connection Failover
- Configuring Simple FAN for JDBC Thin Drvier Non-Pooled Clients
- Configuring JDBC Clients for Application Continuity for Java
- Configuring JDBC-Thin Clients to Use Transaction Guard

> **Note:** When using the Universal Connection Pool (UCP), configure Fast Application Notification and, optionally, Application Continuity. Do not configure Connection Failure Notification (the `ConnectionFailureNotification` connection property).

See Also:

- "About Workload Management with Dynamic Database Services" on page 6-1
- "About Oracle Services" on page 6-2
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about deploying Oracle RAC for high availability
- *Oracle Universal Connection Pool for JDBC Developer's Guide* for more information about the Universal Connection Pool for Java

Configuring JDBC Clients for Fast Connection Failover

The Universal Connection Pool (UCP) subscribes to FAN when Oracle Notification Service is configured and the pool property `setFastConnectionFailoverEnabled` is enabled.

UCP also subscribes to the FAN Load Balancing events if you configure the dynamic database service used in the connection for run-time load balancing. Instead of randomly assigning a free connection to a work request, the connection pool chooses the connection that gives the best service according to the latest information it has received. If a node becomes hung, then run-time load balancing quickly shifts connections from the hung node to other nodes in the cluster.

To configure JDBC clients for Fast Connection Failover:

1. Use the Cluster Managed Services page in Oracle Enterprise Manager Cloud Control to create new services. See "Creating Services" on page 6-17 for more information about creating services.
2. Enable fast connection failover for JDBC clients by setting the UCP DataSource property `FastConnectionFailoverEnabled` to `TRUE`. Also, configure a remote Oracle Notification Services (ONS) subscription within the JDBC application so that the Java Virtual Machine (JVM) can spawn the ONS when needed, as shown in the following example:

```
PoolDataSource pds = PoolDataSourceFactory.getPoolDataSource();
Pds.setConnnectionPoolName("FCFSampleUCP");
pds.setONSConfiguration("nodes=racnode1:4200,racnode2:4200");
pds.setFastConnectionFailoverEnabled(true);
pds.setConnectionFactoryClassName("oracle.jdbc.pool.OracleDataSource");
```

The remote ONS subscription must contain every host that the client application can use for failover.

3. Set the `oracle.net.ns.SQLnetDef.TCP_CONNTIMEOUT_STR` property to a nonzero value on the data source. When this property is set, if the JDBC client attempts to connect to a host that is unavailable, then the connection attempt is bounded to the time specified for `oracle.net.ns.SQLnetDef.TCP_CONNTIMEOUT_STR`. After the specified time has elapsed and a successful connection has not been made, the client attempts to connect to the next host in the address list. Setting this property to a value of three seconds is sufficient for most installations.

4. Configure JDBC clients to use a connect descriptor that includes the SCAN and the service name, and the port that the SCAN listener listens on (if not using the default value of 1521), for example:

```
@//docrac.example.com:1525/orcl_JDBC
```

> **Note:** Do not configure Transparent Application Failover (TAF) with Fast Connection Failover for JDBC clients as TAF processing can interfere with FAN ONS processing.

For example, if the database initialization parameter `remote_listener` is set to the SCAN listener addresses, then, in the JDBC application, you would connect to the database using a connect string similar to the following:

```
pds.setURL("jdbc:oracle:thin:@(DESCRIPTION =
 (TRANSPORT_CONNECT_TIMEOUT=3)(RETRY_COUNT=3)(FAILOVER=ON)(CONNECT_TIMEOUT=60)
 (ADDRESS_LIST =(ADDRESS=(PROTOCOL=tcp)
 (HOST=docrac.example.com)(PORT=1521))
 (CONNECT_DATA=(SERVICE_NAME=service_name)))
```

If you are using a JDBC driver, then you must include the complete connect descriptor in the URL because the JDBC Thin driver does not use Oracle Net.

5. Verify that both `ucp.jar` and `ons.jar` are in the application CLASSPATH.

See Also:

- "Creating Services"
- "About Fast Application Notification (FAN)" on page 6-6
- "Configuring Clients for High Availability" on page 6-23
- *Oracle Database JDBC Developer's Guide* for more information about installing and configuring Oracle RAC Fast Application Notification for Java
- *Oracle Universal Connection Pool for JDBC Developer's Guide* for more information about using the Universal Connection Pool for database connections using Java
- *Oracle Database 2 Day + Java Developer's Guide* for information about creating a method to authenticate users
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about installing and configuring Oracle RAC Fast Application Notification

Configuring Simple FAN for JDBC Thin Drvier Non-Pooled Clients

JDBC simple FAN enables standalone Oracle JDBC connections to quickly respond to node failures. This feature enables a standalone JDBC Thin Driver connection to listen for FAN events and respond to a node down event by marking the connection closed, permitting user applications to begin recovery quicker. It does not alter the behavior of the application, except that JDBC method calls throw exceptions more quickly after a node goes down.

To configure simple FAN for non-pooled JDBC clients, you use the `ConnectionFailureNotification` connection property.

To configure FAN for non-pooled JDBC Clients:

1. Set the Connection Failure Notification property either as a system property using the -D option, or by including it in the connection properties argument when creating a connection.

   To set it as a system property, use a command similar to the following:

   ```
   java -Doracle.jdbc.ConnectionFailureNotification=true
   ```

   To set it within the connection properties argument, use code similar to the following:

   ```
   Properties props = new Properties();
   props.put("ConnectionFailureNotification", "true");
   Connection = DriverManager.getConnection(url, props);
   ```

2. Configure the transport mechanism by which the node down events are received. If ONS is the selected transport mechanism, then use the `SetONSConfiguration` property, as demonstrated in the following code, where `racnode1` and `racnode2` represent nodes in the cluster that have ONS running on them:

   ```
   props.setONSConfiguration("nodes=racnode1:4200,racnode2:4200");
   ```

3. Verify that `ons.jar` is in the application CLASSPATH.

> **See Also:**
>
> - "About Oracle RAC High Availability Framework" on page 6-6
> - "Configuring Clients for High Availability" on page 6-23
> - *Oracle Database JDBC Developer's Guide* for information about installing and configuring Oracle RAC Fast Application Notification
> - *Oracle Database 2 Day + Java Developer's Guide* for information about creating a method to authenticate users

Configuring JDBC Clients for Application Continuity for Java

A new Replay data source (`oracle.jdbc.replay.OracleDataSource`) serves as the front-end for applications to obtain a connection when using Application Continuity for Java. This data source serves as the connection factory that produces new physical JDBC connections, for both UCP and WebLogic Server Active GridLink data sources. In collaboration with Oracle Database, the JDBC replay driver maintains a history of calls during a client conversation with Oracle Database 12*c*, as instructed by the database. Following any outage of the session caused by a loss of database service, planned or unplanned, the JDBC replay driver attempts to rebuild the non-transactional and transactional database session states, so that the outage appears as a delayed execution.

To use Application Continuity for Java and the JDBC replay driver, you must use an Oracle Database 12*c* client and connect to an Oracle Database 12*c* database. Application Continuity for Java is supported in the following configurations:

- JDBC applications using Oracle JDBC Replay data source and using neither UCP or WebLogic Server Active GridLink—typical standalone JDBC cases
- JDBC applications using UCP data sources—standalone or 3rd party application servers configured to use a UCP data source
- JDBC applications using only WebLogic Server Active GridLink but not UCP data sources—typical WebLogic Server J2EE cases

To configure JDBC clients to use the JDBC Replay Driver:

1. Verify you are using an application that has been certified for replay.
2. Use SRVCTL to create a service for use by the application, if one does not already exist. Set the `FAILOVER_TYPE` attribute to `TRANSACTION`, `commit_outcome` attribute to `TRUE`, and notification for FAN to `TRUE` for this service..

   See "Creating Services" for more information about creating services.
3. Configure the connection element using the `replayDataSource` object, as shown in the following example:

   ```
   replayDataSource rds = PoolDataSourceFactory.getreplayDataSource();
   rds.setConnnectionPoolName("replayExample");
   rds.setONSConfiguration("nodes=racnode1:4200,racnode2:4200");
   rds.setFastConnectionFailoverEnabled(true);
   rds.setConnectionFactoryClassName("oracle.jdbc.replay.OracleDataSource");

   Connection conn = replayDataSource.getConnection();
   ```

   Refer to the URL example in "Configuring JDBC Clients for Fast Connection Failover".
4. When connecting to the database, use a URL that can access all instances offering the service. See "Configuring Oracle Net Services Parameters For Highly Available Connections" for more information about creating highly available database connections.

Note: If you disconnect or terminate a session that uses Application Continuity, then Application Continuity attempts to recover the session. To disconnect or terminate the session without replay, see *Oracle Database Development Guide*.

See Also:

- *Oracle Database JDBC Developer's Guide* for information about configuring Oracle JDBC for Application Continuity for Java
- *Oracle Database Development Guide* for more information about using Application Continuity
- *Oracle Universal Connection Pool for JDBC Developer's Guide* for information about using Application Continuity with the Universal Connection Pool

Configuring JDBC-Thin Clients to Use Transaction Guard

Transaction Guard is a reliable protocol and tool that returns the outcome of the last in-flight transaction after an outage that makes the database session unavailable. Transaction Guard avoids the costs of ambiguous errors that lead to user frustration, customer support calls, and lost opportunities. Without Transaction Guard, applications and users who attempt to retry operations following an outage can cause logical corruption by committing duplicate transactions or committing transactions out of order.

See Also:

- *Oracle Database JDBC Developer's Guide* for information about configuring JDBC-Thin clients for use with Transaction Guard
- *Oracle Database Development Guide* for more information about using Transaction Guard

Configuring OCI Clients for High Availability

The Oracle Call Interface (OCI) provides integration with FAN and Load Balancing Advisory events. To take advantage of the Load Balancing Advisory, you must enable the OCI Session pool.

This section explains how to perform the following tasks:

- Configuring OCI Clients to Receive FAN Notifications

See Also:

- "Using Transaction Guard" in *Oracle Database Development Guide* and in OCI Developers Guide
- "Configuring Transparent Application Failover" in *Oracle Call Interface Programmer's Guide*

Configuring OCI Clients to Receive FAN Notifications

OCI clients can register to receive notifications about Oracle RAC high availability events and respond when events occur. This improves the connection failover response time in OCI and also removes terminated connections from connection and session pools. This feature works for all OCI client applications.

To configure OCI clients to receive FAN notifications:

1. Use the Cluster Managed Services page in Oracle Enterprise Cloud Control to a create services for the OCI clients. See "Creating Services" on page 6-17 for more information about creating services.

   You should configure the primary instance as preferred for that service. For Notification Properties, choose "Enable Fast Application Notification for OCI and ODP.NET Applications".

   See "Administering Services" on page 6-21 for more information about modifying services using Enterprise Manager.

2. Link the OCI client applications with thread library `libthread` or `libpthread`.

3. In your application, you must check if an event has occurred, using code similar to the following example:

   ```
   void evtcallback_fn(ha_ctx, eventhp)
   ...
   ```

```
printf("HA Event received.\n");
  if (OCIHandleAlloc( (dvoid *)envhp, (dvoid **)&errhp, (ub4) OCI_HTYPE_ERROR,
                      (size_t) 0, (dvoid **) 0))
    return;
  if (retcode = OCIAttrGet(eventhp, OCT_HTYPE_EVENT, (dvoid *)&srvhp, (ub4 *)0,
                           OCI_ATTR_HA_SRVFIRST, errhp))
    checkerr (errhp, (sword)retcode;
  else {
     printf("found first server handle.\n");
     /*get associated instance name */
     if (retcode = OCIAttrGet(srvhp, OCI_HTYPE_SERVER, (dvoid *)&instname,
                           (ub4 *)&sizep, OCI_ATTR_INSTNAME, errhp))
       checkerr (errhp, (sword)retcode);
     else
       printf("instance name is %s.\n", instname);
```

4. After a HA event is received, clients and applications can register a callback that is invoked whenever a high availability event occurs, as shown in the following example:

```
/*Registering HA callback function */
  if (checkerr(errhp, OCIAttrSet(envhp, (ub4) OCI_HTYPE_ENV,
                             (dvoid *)evtcallback_fn, (ub4) 0,
                             (ub4)OCI_ATTR_EVTCBK, errhp)))
  {
    printf("Failed to set register EVENT callback.\n");
    return EX_FAILURE;
  }
  if (checkerr(errhp, OCIAttrSet(envhp, (ub4) OCI_HTYPE_ENV,
                                (dvoid *)evtctx, (ub4) 0,
                                (ub4)OCI_ATTR_EVTCTX, errhp)))
  {
    printf("Failed to set register EVENT callback context.\n");
    return EX_FAILURE;
  }
return EX_SUCCESS;
```

After registering an event callback and context, OCI calls the registered function once for each high availability event.

See Also:

- "Creating Services"
- "Administering Services"
- "About Fast Application Notification (FAN)" on page 6-6
- "Configuring Clients for High Availability" on page 6-23
- *Oracle Call Interface Programmer's Guide* for more information about event notification and user-registered callbacks
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring fast application notification for OCI clients.

Configuring ODP.NET Clients for High Availability

Oracle Data Provider for .NET (ODP.NET) connection pools subscribe to FAN notifications from Oracle RAC that indicate when nodes are down and when services are up or down. Based on these notifications, the ODP.NET connection pool:

- Interrupts active sessions so they do not wait on TCP/IP timeouts, and
- Clears idle sessions previously connected to the down node or instance or network so they are not passed to user sessions
- Creates new connections to healthy nodes, if possible

ODP.NET provides Run-time Connection Load Balancing to provide enhanced load balancing of the application workload. Instead of randomly selecting an available connection from the connection pool, it chooses the connection that can provide the best service based on the current workload information.

- Configuring ODP.NET Clients to Receive FAN Notifications
- About ODP.NET Clients and Using Transaction Guard

Configuring ODP.NET Clients to Receive FAN Notifications

The procedures for enabling ODP.NET are similar to the procedures for enabling JDBC in that you must set parameters in the connection string to enable FCF. This section explains how to configure Oracle Data Provider for .NET (ODP.NET) clients for failover using FAN events.

To configure ODP.NET clients to receive FAN notifications:

1. Use the Cluster Managed Services page in Oracle Enterprise Manager Cloud Control to a create services for the ODP.NET clients. See "Creating Services" on page 6-17 for more information about creating services.

   For Notification Properties, choose "Enable Fast Application Notification for OCI and ODP.NET Applications". Set the Connection Load Balancing Goal to Long.

2. Enable Fast Connection Failover for ODP.NET connection pools by subscribing to FAN high availability events. Do this by setting the `ha events` connection string attribute to `true` either at connection time or in the data source definition. Note that this only works if you are using connection pools (the `pooling` attribute to `true`).

   You can also enable Run-time Connection Load Balancing by setting the run-time load balancing attribute for the service to SERVICE_TIME.

   Use code similar to the following, where *username* is the name of the database user that the application uses to connect to the database, *password* is the password for that database user, and the service name is `odpserv`:

```
// C#
using System;
using Oracle.DataAccess.Client;
class HAEventEnablingSample
{
  static void Main()
  {
    OracleConnection con = new OracleConnection();

    // Open a connection using ConnectionString attributes
    // Also, enable "load balancing"
    con.ConnectionString =
          "User Id=username;Password=password;" +
          "Data Source=//docrac.example.com/odpserv;" +
          "Min Pool Size=10;Connection Lifetime=120;Connection Timeout=60;" +
          "HA Events=true;Incr Pool Size=5;Decr Pool Size=2";

        con.Open();
        // Carry out work against the database here.
        con.Close();
        // Dispose OracleConnection object
        con.Dispose();
        }
     }
```

3. The run-time load balancing events are published as messages in the SYS.SYS$SERVICE_METRICS queue. You must grant dequeue permission on this queue to the database user that the application uses to connect to the database.

   Use a command similar to the following, where *username* represents the database user that the .NET application uses to connect to the database:

   ```
   execute
   dbms_aqadm.grant_queue_privilege('DEQUEUE','SYS.SYS$SERVICE_METRICS',
   username);
   ```

   The *username* specified in this step equals the *username* used for the User Id argument in the previous step.

   See Also:

   - "Creating Services"
   - "About Fast Application Notification (FAN)" on page 6-6
   - "Configuring Clients for High Availability" on page 6-23
   - *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows* for more information about event notification and user-registered callbacks
   - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring fast application notification for ODP.NET clients.

About ODP.NET Clients and Using Transaction Guard

When an ODP.NET application submits a transaction commit request and some type of failure occurs, such as a node, network, or database failure, ODP.NET can now deterministically conclude whether the transaction committed or not. Starting with Oracle Database 12*c*, applications can use a logical transaction ID (LTXID) to determine the outcome of the last transaction open in a database session following an outage.

Using Transaction Guard, the end user experience is vastly improved by returning to the application and user, following an outage, whether the last submission committed and completed or did not. A committed result stays committed. An uncommitted result stays uncommitted, and is a green light, for example, for the user or application to resubmit.

Without using Transaction Guard, applications that attempt to retry operations following outages can cause logical corruption by committing duplicate transactions.

See Also:

- "Creating Services"
- "Administering Services"
- *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows* for an example of using Transaction Guard in an ODP.NET application
- "About Fast Application Notification (FAN)" on page 6-6
- "Configuring Clients for High Availability" on page 6-23
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring fast application notification for ODP.NET clients.

7 Administering Backup and Recovery

To protect the database against data loss and reconstruct the database after data loss, you must devise, implement, and manage a backup and recovery strategy for your Oracle RAC Database.

See Also:

- *Oracle Database 2 Day DBA* for conceptual information about database backup and recovery
- *Oracle Database Backup and Recovery User's Guide* for more information about using the Recovery Manager utility

Overview of Oracle RAC Database Backup and Recovery

To protect your Oracle Real Application Clusters (Oracle RAC) database from hardware failures or disasters, you must have a physical copy of the database files. The files protected by the backup and recovery facilities built into Oracle Enterprise Manager include data files, control files, server parameter files (SPFILEs), and archived redo log files. Using these files, your database can be reconstructed. The backup mechanisms that work at the physical level protect against damage at the file level, such as the accidental deletion of a data file or the failure of a disk drive. **Database recovery** involves restoring, or copying, the damaged files from backup and performing media recovery on the restored files. **Media recovery** is the application of redo logs or incremental backups to a restored data file to update it to the current time or some other specified time.

The Oracle Database flashback features, such as Oracle Flashback Drop and Oracle Flashback Table, provide a range of physical and logical data recovery tools as efficient, easy-to-use alternatives to physical and logical backup operations. The flashback features enable you to reverse the effects of unwanted database changes without restoring data files from backup or performing media recovery.

The Enterprise Manager physical backup and recovery features are built on the Recovery Manager (RMAN) command-line client. Enterprise Manager makes available many of the RMAN features, and provides wizards and automatic strategies to simplify and further automate RMAN-based backup and recovery.

> **Note:** For the RMAN utility to work properly on Linux platforms, the `$ORACLE_HOME/bin` directory must appear in the `PATH` variable before the `/usr/X11R6/bin` directory.

The Enterprise Manager Guided Recovery capability provides a Recovery Wizard that encapsulates the logic required for a wide range of file restoration and recovery scenarios, including the following:

- Complete restoration and recovery of the database
- Point-in-time recovery of the database or selected tablespaces
- Flashback Database
- Other flashback features of Oracle Database for logical-level repair of unwanted changes to database objects
- Media recovery at the block level for data files with corrupt blocks

If the database files are damaged or need recovery, then Enterprise Manager can determine which parts of the database must be restored from a backup and recovered, including early detection of situations such as corrupted database files. Enterprise Manager guides you through the recovery process, prompting for needed information and performing the required recovery actions.

See Also:

- "Performing Backups of Your Oracle Real Application Clusters Database"
- "Recovering Your Oracle Real Application Clusters Database"
- "About Managing Your Database Backup Files"
- *Oracle Database 2 Day DBA* for information about performing Oracle Database backup and recovery with Oracle Recovery Manager (RMAN)

About the Fast Recovery Area in Oracle RAC

Using a fast recovery area minimizes the need to manually manage disk space for your backup-related files and balance the use of space among the different types of files. Oracle recommends that you enable a fast recovery area to simplify your backup management.

The larger the fast recovery area is, the more useful it becomes. Ideally, the fast recovery area should be large enough to contain all the following files:

- A copy of all data files
- Incremental backups
- Online redo logs

- Archived redo log files that have not yet been backed up
- Control files and control file copies
- Autobackups of the control file and database initialization parameter file

The fast recovery area for an Oracle RAC database must be placed on an Oracle ASM disk group, a cluster file system, or on a shared directory that is configured through a network file system file for each Oracle RAC instance. In other words, the fast recovery area must be shared among all of the instances of an Oracle RAC database.

> **Note:** The preferred configuration for Oracle RAC is to use Oracle Automatic Storage Management (Oracle ASM) for storing the fast recovery area, using a different disk group for your recovery set than for your data files.

The location and disk quota must be the same on all instances. Oracle recommends that you place the fast recovery area on the shared Oracle ASM disks. In addition, you must set the `DB_RECOVERY_FILE_DEST` and `DB_RECOVERY_FILE_DEST_SIZE` parameters to the same values on all instances.

To use the fast recovery feature, you must first configure the fast recovery area for each instance in your Oracle RAC database.

See Also:

- "Oracle Automatic Storage Management in Oracle RAC"
- "Creating Additional Oracle ASM Disk Groups" on page 3-11
- "Editing Initialization Parameter Settings for an Oracle RAC Database" on page 4-8
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*
- *Oracle Database Backup and Recovery User's Guide* for information about configuring the fast recovery area

Archiving the Oracle Real Application Clusters Database Redo Logs

To make your data highly available, it is important to configure the database so you can recover your data after a system failure. Redo logs contain a record of changes that were made to datafiles. Redo logs are stored in redo log groups, and you must have at least two redo log groups for your database.

After the redo log files in a group have filled up, the log writer process (LGWR) switches the writing of redo records to a new redo log group. Oracle Database can automatically save the inactive group of redo log files to one or more offline destinations, known collectively as the **archived redo log** (also called the archive log). The process of turning redo log files into archived redo log files is called **archiving**.

When you archive your redo log, you write redo log files to another location before they are overwritten. This location is called the archived redo log. These copies of redo log files extend the amount of redo data that can be saved and used for recovery. Archiving can be either enabled or disabled for the database, but Oracle recommends that you enable archiving.

This section contains the following topics:

- About Archived Redo Log Files for an Oracle RAC Database
- About Parallelism and Backups Across Multiple RMAN Channels
- Configuring Archiving for Your Oracle RAC Database

See Also:

- "Administering Redo Logs in Oracle RAC"
- "Oracle Automatic Storage Management in Oracle RAC"
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*

About Archived Redo Log Files for an Oracle RAC Database

When you use Oracle Database Configuration Assistant (DBCA) to create your Oracle Real Application Clusters (Oracle RAC) database, each instance is configured with at least two redo log files that are stored in the shared storage area. If you have a two-node Oracle RAC database, then at least four redo logs are created for the database, two for each instance.

If you use a cluster file system to store the archived redo log files for your Oracle RAC database, then the redo log files are shared file system files. If you use Oracle ASM to store the archived redo log files for your Oracle RAC database, then each instance automatically has access to all the archived redo log files generated by the database. If you use shared storage to store the archived redo log files on each node, then you must configure the operating system to grant access to those directories for each instance of the cluster database that needs access to them.

The primary consideration when configuring archiving is to ensure that all archived redo logs can be read from every node during recovery, and if possible during backups. During recovery, because the archived log destinations are visible from the node that performs the recovery, Oracle RAC can successfully recover the archived redo log data. For creating backups of your Oracle RAC database, the strategy that you choose depends on how you configure the archiving destinations for each node. Whether only one node or all nodes perform archived redo log backups, you must ensure that the archived redo logs for every instance are backed up.

To backup the archived redo logs from a single node, that node must have access to the archived log files of the other instances. The archived redo log naming scheme that you use is important because when a node writes to a log with a specific file name on its file system, the file must be readable by any node that must access this archived redo log. For example, if `node1` archives a log to `/oracle/arc_dest/log_1_100_23452345.arc`, then `node2` can back up this archived redo log only if it can read `/oracle/arc_dest/log_1_100_23452345.arc` on its own file system.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about creating archived log files in Oracle ASM
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about naming conventions for archived redo log files in Oracle RAC

About Parallelism and Backups Across Multiple RMAN Channels

Recovery Manager (RMAN) depends on server sessions, processes that run on the database server, to perform backup and recovery tasks. Each server session in turn corresponds to an RMAN channel, representing one stream of data to or from a backup device. RMAN supports parallelism, which is the use of multiple channels and server sessions to perform the work of a single backup job or file restoration task.

Because the Oracle Database control file, SPFILE, and data files are accessible by any instance, the backup operation of these files is distributed across all the allocated channels. For backups of the archived redo log, the actions performed by RMAN depend on the type of archiving scheme used by your Oracle RAC database.

If you use a local archiving scheme, then each instance writes the archived redo log files to a local directory. When multiple channels are allocated that have access to the archived redo log, for each archived redo log file, RMAN determines which channels have access to that archived redo log file. Then, RMAN groups the archived redo log files that can be accessed by a channel and schedules a backup job using that channel.

If each node in the cluster writes the archived redo log files to Oracle ASM, a clustered file system, or other type of shared storage, then each instance has access to all the archived redo log files. In this case, the backup of the archived redo log is distributed across all the allocated channels.

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- "Performing Backups of Your Oracle Real Application Clusters Database"
- *Oracle Database 2 Day DBA* for more information about RMAN backups
- *Oracle Database 2 Day DBA* for more information about configuring backup device settings

Configuring Archiving for Your Oracle RAC Database

For Oracle RAC, each instance has its own thread of redo. The preferred configuration for Oracle RAC is to configure the fast recovery area using an Oracle ASM disk group that is separate from the Oracle ASM disk group used for your data files. Alternatively, you can use a cluster file system archiving scheme.

To configure archiving for your Oracle RAC database:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Backup and Recovery**, and then **Recovery Settings**.

   The Recovery Settings page appears.

3. In the Media Recovery section, select the **ARCHIVELOG mode** option.

4. In the **Log Archive Filename Format** field, accept the default value, or enter the desired format.

For clustered databases, the format for the archive log file name should contain the %t modifier, to indicate which redo log thread the archived redo log file belongs to. As a best practice, the format for the archive log file name should also include the %s (log sequence number) and %r (resetlogs identifier) modifiers.

5. If the archive log destination is the same for all instances, then in the **Archive Log Destination** field, change the value to the location of the archive log destination for the cluster database.

   For example, you might set it to +DATA if using Oracle ASM, or to /u01/oradata/arch if you want local archiving on each node.

   If you must configure a different archive log destination for any instance, then you must go to the Initialization Parameters page and modify the LOG_ARCHIVE_DEST_1 parameter that corresponds to the instance for which you want to configure the archive log destination. The Instance column should display the name of the instance, for example sales1. Change the **Value** field to contain the location of the archive log destination for that instance.

6. If you want to configure multiple archive log destinations for the database, then on the Recovery Settings page, click **Add Another Row** under the Archive Log Destination field.

7. After you have finished configuring archiving, click **Apply**.

   When prompted to restart the database, click **Yes**.

8. Enter the host and SYSDBA user credentials, then click **Continue**.

9. Wait a couple of minutes, then click **Refresh**.

   If the database has been restarted, then you are prompted to enter the login credentials.

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- "About Configuring Initialization Parameters for an Oracle RAC Database"
- "Editing Initialization Parameter Settings for an Oracle RAC Database"
- *Oracle Database 2 Day DBA* for information about configuring settings used for instance recovery, media recovery, and fast recovery
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring and managing archived redo log files for an Oracle RAC database

About Preparing for Backup and Recovery Operations

Before taking backups of your Oracle Real Application Clusters (Oracle RAC) database using Enterprise Manager, you must configure access for the user performing the backups, or **credentials**. You can also configure the default values for certain backup settings, so they do not have to be specified every time a backup is taken.

See Also:

- "About Operating System Users and Groups"

About Credentials for Performing Backup and Recovery

When using Enterprise Manager, you must have the proper credentials to perform the configuration tasks for backup and recovery, to schedule backup jobs, and to perform recovery. The following credentials may be required:

- The Oracle database administrator user you use when you log in to Enterprise Manager
- The host operating system user whose credentials you provide when performing backup and recovery tasks

To perform or schedule RMAN tasks, you must either log in to Enterprise Manager as a user with `SYSDBA` privileges, or provide host operating system credentials for a user who is a member of the `dba` group. The host operating system user must also have execute permission for the RMAN command-line client.

For tasks requiring host operating system credentials, a Host Credentials form appears at the bottom of the page used to perform the task. Enterprise Manager uses the credentials when it invokes RMAN to perform jobs you requested or scheduled.

The Host Credentials form always includes an option labeled Save as Preferred Credential. If you select this option before performing your action, then the provided credentials are stored persistently for the currently logged-in Oracle database user. The preferred credentials are reused by default whenever you log in as that user and perform operations requiring host credentials.

See Also:

- "About Operating System Users and Groups"
- "About Configuring User Accounts"
- *Oracle Database 2 Day DBA* for information about the user credentials required to perform backup and recovery

About Configuring Backup Settings

Assuming you have a fast recovery area configured, you can configure several settings and policies that determine how backups are stored, which data is backed up, and how long backups are retained before being purged from the fast recovery area. You can also configure settings to optimize backup performance for your environment.

See Also:

- "Configuring Archiving for Your Oracle RAC Database"
- "Overview of Oracle RAC Database Backup and Recovery"
- *Oracle Database 2 Day DBA* for more information about configuring backup policy settings
- *Oracle Database 2 Day DBA* for more information about configuring backup settings

Performing Backups of Your Oracle Real Application Clusters Database

When you use Oracle ASM to manage database files, Oracle recommends that you use RMAN for creating backups. You must have both database (`SYSDBA`) privileges and host operating system (`OSDBA`) credentials to perform backup and recovery operations.

If you log in to Enterprise Manager with `SYSDBA` privileges, then any operating system user who has execute permission for the RMAN command-line client can perform backups of an Oracle Real Application Clusters (Oracle RAC) database. However, if you log in as a database user without `SYSDBA` privileges, then you must provide the name and password of an operating system user that is a member of the `OSDBA` group before you can perform the backup operation.

To back up an Oracle RAC database:

1. On the Cluster Database Home page, select **Availability**.

   The Cluster Database Availability page appears.

2. In the Backup/Recovery section, under the heading Manage, select **Schedule Backup**.
3. Follow the backup procedures outlined in Chapter 9, "Performing Backup and Recovery" of *Oracle Database 2 Day DBA* or click **Help** on this page.

   See Also:

   - "About Operating System Users and Groups"
   - "About Credentials for Performing Backup and Recovery"
   - *Oracle Database 2 Day DBA* for more information about configuring your database for backup and recovery
   - *Oracle Database 2 Day DBA* for more information about performing and scheduling backups using RMAN

Backing Up Archived Redo Log Files

Whether only one node or all nodes perform archive log backups, ensure that all archived redo log files for all nodes are backed up. If you use a local archiving scheme, then allocate multiple channels to provide RMAN access to all the archived redo log files.

You can configure RMAN to automatically delete the archived redo log files from disk after they have been safely backed up. This feature helps to reduce the disk space used by your Oracle RAC database, and prevents unnecessary outages that might occur if you run out of available disk space.

To configure RMAN to automatically delete the archived redo log file files from disk after they have been safely backed up, when creating or scheduling your database backups:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Backup and Recovery**, and then **Schedule Backup**.
3. Choose a backup type and click **Schedule Customized Backup**.
4. While specifying the options for your backup, select **Also back up all archived logs** on disk if you are performing an online backup. There is no need to back up archived redo log files when performing an offline backup because the database is in a consistent state at the time of backup and does not require media recovery if you restore.

5. Select **Delete all archived logs from disk after they are successfully backed up** if you are using shared storage for your archived redo log files.

Note: Do not select **Delete all archived logs from disk after they are successfully backed up** if you are using a fast recovery area as your *only* archive log destination. In this case, archived redo log files that have been backed up are deleted automatically as space is needed for storage of other files.

See Also:

- "Performing Backups of Your Oracle Real Application Clusters Database"
- "About Archived Redo Log Files for an Oracle RAC Database"
- *Oracle Database 2 Day DBA* for more information about configuring backup settings

Recovering Your Oracle Real Application Clusters Database

The Enterprise Manager Guided Recovery capability provides a Recovery Wizard that encapsulates the logic required for a wide range of restore and recovery scenarios. Enterprise Manager can determine which parts of the database must be restored and recovered, including early detection of situations such as corrupted database files. Enterprise Manager takes you through the recovery process, prompting for information and performing required file restoration and recovery actions.

This section discusses both instance recovery and media recovery. It contains the following topics:

- About Preparing to Restore and Recover Your Oracle RAC Database
- About Restoring Archived Redo Log File Files
- About Performing Recovery Using Parallelism
- Recovering the Oracle RAC Database Using Enterprise Manager
- Recovering the Server Parameter File from an Automatic File Backup

See Also:

- "About Oracle Real Application Clusters"
- "Overview of Oracle RAC Database Backup and Recovery"
- "Archiving the Oracle Real Application Clusters Database Redo Logs"
- "About the Fast Recovery Area in Oracle RAC"
- "Performing Backups of Your Oracle Real Application Clusters Database"

About Preparing to Restore and Recover Your Oracle RAC Database

The node that performs the recovery of an Oracle Real Application Clusters (Oracle RAC) database must be able to restore all the required data files. That node must also be able to either read all the required archived redo log files on disk or be able to restore the archived redo log files from backup files.

This section discusses two tasks you must perform before recovering your database:

- About Configuring Access to the Archive Log
- About Putting the Oracle RAC Database Instances into the Correct State

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- "Configuring Shared Storage"
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*

About Configuring Access to the Archive Log

During recovery, because the archive log file destinations are visible from the node that performs the recovery, Oracle RAC can successfully access the archived redo log files during recovery.

If you do not use shared storage or a clustered file system to store the archived redo log files for your cluster database, then you must make the archived redo log files available to the node performing the recovery.

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- "Administering Redo Logs in Oracle RAC"
- "About Preparing to Restore and Recover Your Oracle RAC Database"
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*

About Putting the Oracle RAC Database Instances into the Correct State

Recovery of a failed instance in Oracle RAC is automatic. If an Oracle RAC database instance fails, then a surviving database instance processes the online redo logs generated by the failed instance to ensure that the database contents are in a consistent state. When recovery completes, Oracle Clusterware attempts to restart the failed instance automatically.

Media recovery is a manual process that occurs while a database is closed. A media failure is the failure of a read or write operation of a disk file required to run the database, due to a physical problem with the disk such as a head malfunction. Any database file can be vulnerable to a media failure. If a media failure occurs, then you must perform media recovery to restore and recover the damaged database files. Media recovery is always done by one instance in the cluster.

Before starting media recovery, the instance that is performing the recovery should be started in `MOUNT` mode. The other instances should be started in `NOMOUNT` mode.

See Also:

- "Starting and Stopping Oracle RAC Databases and Database Instances"
- "About Preparing to Restore and Recover Your Oracle RAC Database"
- *Oracle Database 2 Day DBA* for information about the Oracle advised recovery feature and Data Recovery Advisor

About Restoring Archived Redo Log File Files

During a restore operation, RMAN automatically locates the most recent backups of the database that are available. A channel connected to a specific node attempts to restore files that were backed up only to that node. For example, assume that an archived redo log file with the sequence number 1_001 is backed up to a device attached to the node `racnode1`, while the archived redo log file with sequence number 2_003 is backed up to a device attached to the node `racnode2`. If you allocate channels that connect to nodes `racnode1` and `racnode2` for a restore operation, then the channel connected to `racnode1` restores log sequence 1_001, but not log sequence 2_003. The channel connected to `racnode2` can restore log sequence 2_003, but not log sequence 1_001.

If you use Oracle ASM or a clustered file system for storing the archived redo log files, then any instance can restore the archived redo log files.

See Also:

- "About Archived Redo Log Files for an Oracle RAC Database"
- "Configuring Archiving for Your Oracle RAC Database"
- *Oracle Database Backup and Recovery User's Guide* for more information about restoring archived redo log file files

About Performing Recovery Using Parallelism

Oracle RAC automatically selects the optimum degree of parallelism for instance failure and media recovery.

When using Enterprise Manager and RMAN to perform the recovery, Oracle RAC automatically makes parallel the following three stages of recovery:

- Restoring Data files—When restoring data files, the number of channels you allocate in the RMAN recovery script effectively sets the parallelism that RMAN uses. For example, if you allocate five channels, then you can have up to five parallel streams restoring data files.
- Applying Incremental Backups—Similarly, when you are applying incremental backups, the number of channels you allocate determines the potential parallelism.
- Applying Archived Redo Log Files—Using RMAN, the application of archived redo log files is performed in parallel. Oracle RAC automatically selects the optimum degree of parallelism based on available CPU resources.

See Also:

- "Recovering Your Oracle Real Application Clusters Database"
- "Overview of Oracle RAC Database Backup and Recovery"
- *Oracle Database 2 Day DBA* for more information about incremental backups of data files
- *Oracle Database 2 Day DBA* for more information about configuring recovery settings

Recovering the Oracle RAC Database Using Enterprise Manager

When using Enterprise Manager and RMAN, the process of recovering and restoring an Oracle RAC database is essentially the same as for a single-instance Oracle database, except that you access RMAN from the Availability page at the cluster database level, instead of at the instance level.

To use Enterprise Manager and RMAN to restore and recover an Oracle RAC database:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Backup and Recovery**, and then **Perform Recovery**.

   The Perform Recovery page appears.

3. Follow the recovery procedures outlined in Chapter 9 of *Oracle Database 2 Day DBA*.

See Also:

- "About Preparing to Restore and Recover Your Oracle RAC Database"
- "About Credentials for Performing Backup and Recovery"
- *Oracle Database 2 Day DBA* for more information about performing user-directed recovery

Recovering the Server Parameter File from an Automatic File Backup

You can use Enterprise Manager to recover a lost or damaged server parameter file (SPFILE).

To recover an SPFILE for an Oracle RAC database:

1. Start the database in `MOUNT` mode.

2. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

3. From the **Availability** menu, select **Backup and Recovery**, and then **Perform Recovery**.

When the database is not open, the Perform Recovery link takes you to the SPFILE restore page.

4. Specify the location of the fast recovery area, if configured.
5. In the Backup Information section, select **Use Other Backup Information** and **Use an Autobackup**.
6. On the Perform Recovery: Restore SPFILE page, specify a different location for the SPFILE to be restored to.
7. When finished selecting your options, click **Restore**, then click **Yes** to confirm you want to restore the SPFILE.
8. After the SPFILE is restored, you are prompted to log in to the database again.

See Also:

- "Starting and Stopping Oracle RAC Databases and Database Instances"
- "About the Server Parameter File for Oracle Real Application Clusters"
- *Oracle Database Backup and Recovery User's Guide* for more information about recovering a server parameter file

About Managing Your Database Backup Files

Managing RMAN backup files, with or without Enterprise Manager, consists of two tasks:

1. Managing the backup files for your database that are stored on disk or tape
2. Managing the record of those backup files in the RMAN repository

Enterprise Manager simplifies both backup file management tasks. Other tasks involved in managing backup files include the following:

- Searching for backup files
- Validating the contents of backup sets or image copies
- Cross-checking a backup
- Deleting expired or obsolete backup files
- Marking backup files as available or unavailable

See Also:

- "Overview of Oracle RAC Database Backup and Recovery"
- "Performing Backups of Your Oracle Real Application Clusters Database"
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day DBA* for more information about managing RMAN backup files

Displaying Backup Reports for Your Oracle Real Application Clusters Database

Backup reports contain summary and detailed information about past backup jobs run by RMAN, including backup jobs run through Enterprise Manager and the RMAN command-line client.

To view backup reports:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page, and log in as a SYSDBA user.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Backup and Recovery**, and then **Backup Reports**.

   The View Backup Report page appears, with a list of recent backup jobs.

3. In the Search section, specify any filter conditions and click **Go** to restrict the list to backups of interest.

   You can use the **Search** section of the page to restrict the backups listed by the time of the backup, the type of data backed up, and the status of the jobs (whether it succeeded or failed, and whether warnings were generated during the job).

4. To view detailed information about any backup, click the backup job name in the **Backup Name** column.

   The Backup Report page is displayed for the selected backup job. This page contains summary information about this backup job, such as how many files of each type were backed up, the total size of the data backed up, and the number, size, and type of backup files created.

   The Backup Report page also contains a Search section that you can use to quickly run a search for another backup job or backup jobs from a specific date range. The resulting report contains aggregate information for backup jobs matching the search criteria.

   See Also:

   - "About Managing Your Database Backup Files"
   - "Performing Backups of Your Oracle Real Application Clusters Database"
   - "Overview of Oracle RAC Database Backup and Recovery"
   - *Oracle Database 2 Day DBA* for more information about displaying backup reports

8

Monitoring Performance and Troubleshooting

Performance tuning for an Oracle Real Application Clusters (Oracle RAC) database is very similar to performance tuning for a single-instance database. Many of the tuning tasks that you perform on single-instance Oracle databases can also improve performance of your Oracle RAC database. The following topics focus on the performance tuning and monitoring tasks that are unique to Oracle RAC. For information about general performance tuning, refer to *Oracle Database 2 Day DBA*.

See Also:

- *Oracle Database 2 Day DBA* for more information about basic database tuning
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about general performance tuning
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about diagnosing problems for Oracle Real Application Clusters components
- *Oracle Clusterware Administration and Deployment Guide* for more information about diagnosing problems for Oracle Clusterware components

Monitoring Oracle RAC Database and Cluster Performance

Oracle Enterprise Manager Cloud Control is cluster-aware and provides a central console to manage your cluster database.

From the Cluster Database Home page, you can do all of the following:

- View the overall system status, such as the number of nodes in the cluster and their current status. This high-level view capability means that you do not have to access each individual database instance for details if you just want to see inclusive, aggregated information.
- View alert messages aggregated across all the instances with lists for the source of each alert message. An **alert message** is an indicator that signifies that a particular

metric condition has been encountered. A **metric** is a unit of measurement used to report the system's conditions.

- Review issues that are affecting the entire cluster and those that are affecting individual instances.
- Monitor cluster cache coherency statistics to help you identify processing trends and optimize performance for your Oracle RAC environment. Cache coherency statistics measure how well the data in caches on multiple instances is synchronized. If the data caches are completely synchronized with each other, then reading a memory location from the cache on any instance returns the most recent data written to that location from any cache on any instance.
- Determine if any of the services for the cluster database are having availability problems. A service is considered to be a problem service if it is not running on all preferred instances, if its response time thresholds are not met, and so on. Clicking on the link on the Cluster Database Home page opens the Cluster Managed Database services page where the service can be managed.
- Review any outstanding Clusterware interconnect alerts.

Also note the following points about monitoring Oracle RAC environments:

- Performance monitoring features, such as Automatic Workload Repository (AWR) and Statspack, are Oracle RAC-aware.

> **Note:** Instead of using Statspack, Oracle recommends that you use the more sophisticated management and monitoring features of the Oracle Database 12*c* Diagnostic and Tuning packs, which include AWR.

- You can use global dynamic performance views, or GV$ views, to view statistics across instances. These views are based on the single-instance V$ views.

The following topics explain how to monitor both types of data:

- About Automatic Database Diagnostic Monitor and Oracle RAC Performance
- Viewing ADDM for Oracle RAC Findings
- Using the Cluster Database Performance Page

See Also:

- "About Oracle RAC Management Using Enterprise Manager"
- *Oracle Database 2 Day DBA*
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database 2 Day + Performance Tuning Guide*

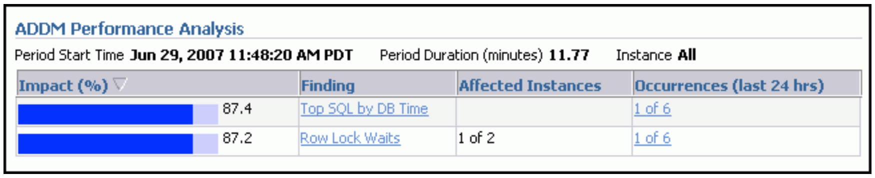
About Automatic Database Diagnostic Monitor and Oracle RAC Performance

The Automatic Database Diagnostic Monitor (ADDM) is a self-diagnostic engine built into the Oracle Database. ADDM examines and analyzes data captured in the Automatic Workload Repository (AWR) to determine possible performance problems in Oracle Database. ADDM then locates the root causes of the performance problems, provides recommendations for correcting them, and quantifies the expected benefits. ADDM analyzes the AWR data for performance problems at both the database and the instance level.

An ADDM analysis is performed as each AWR snapshot is generated, which is every hour by default. The results are saved in the database and can be viewed by using Enterprise Manager. Any time you have a performance problem, you should first review the results of the ADDM analysis. An ADDM analysis is performed from the top down, first identifying symptoms, then refining the analysis to reach the root causes, and finally providing remedies for the problems.

For the cluster-wide analysis, Enterprise Manager reports two types of findings:

- Database findings: An issue that concerns a resource shared by all instances in the cluster database, or an issue that affects multiple instances. An example of a database finding is I/O contention on the disk system used for shared storage.
- Instance findings: An issue that concerns the hardware or software that is available for only one instance, or an issue that typically affects just a single instance. Examples of instance findings are high CPU load or sub-optimal memory allocation.



ADDM Performance Analysis

Period Start Time **Jun 29, 2007 11:48:20 AM PDT** Period Duration (minutes) **11.77** Instance **All**

Impact (%) ▽	Finding	Affected Instances	Occurrences (last 24 hrs)
87.4	Top SQL by DB Time		1 of 6
87.2	Row Lock Waits	1 of 2	1 of 6

ADDM reports only the findings that are significant, or findings that take up a significant amount of instance or database time. Instance time is the amount of time spent using a resource due to a performance issue for a single instance and database time is the sum of time spent using a resource due to a performance issue for all instances of the database, excluding any Oracle Automatic Storage Management (Oracle ASM) instances.

An instance finding can be reported as a database finding if it relates to a significant amount of database time. For example, if one instance spends 900 minutes using the CPU, and the sum of all time spent using the CPU for the cluster database is 1040 minutes, then this finding would be reported as a database finding because it takes up a significant amount of database time.

A problem finding can be associated with a list of recommendations for reducing the impact of the performance problem. Each recommendation has a benefit that is an estimate of the portion of database time that can be saved if the recommendation is implemented. A list of recommendations can contain various alternatives for solving the same problem; you do not have to apply the recommendations.

Recommendations are composed of actions and rationales. You must apply all the actions of a recommendation to gain the estimated benefit of that recommendation. The rationales explain why the actions were recommended, and provide additional information to implement the suggested recommendation.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "About Workload Management with Dynamic Database Services"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about configuring and using AWR and ADDM
- *Oracle Database Performance Tuning Guide* for more information about Automatic Database Diagnostic Monitor

Viewing ADDM for Oracle RAC Findings

By default, Automatic Database Diagnostic Monitor (ADDM) runs every hour to analyze snapshots taken by the AWR during that period. If the database finds performance problems, then it displays the results of the analysis under Diagnostic Summary on the Cluster Database Home page. The ADDM Findings link shows how many ADDM findings were found in the most recent ADDM analysis.

ADDM for Oracle RAC can be accessed in Enterprise Manager by the following methods:

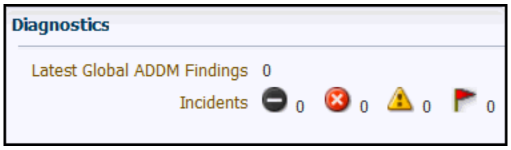
- On the Cluster Database Home Page, in the Diagnostics section, click the **Latest Global ADDM Findings** link.
- On the Cluster Database Performance Home page, click the camera icons at the bottom of the Active Sessions Graph.
- From the **Performance** menu, select **Advisors**, and then either **Advisor Central** or **ADDM**.

To view ADDM findings from the Cluster Database Home page:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. On the Cluster Database Home page, under Diagnostic Summary, if a nonzero number is displayed next to Latest Global ADDM Findings, then click this link.

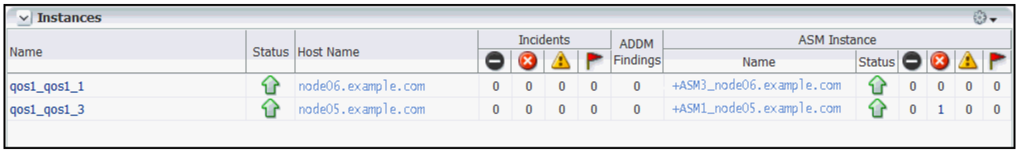


Diagnostics

Latest Global ADDM Findings 0

Incidents ⊖ 0 ⊗ 0 ⚠ 0 🚩 0

   You can also view the ADDM findings per instance by viewing the Instances table on the Cluster Database Home page.

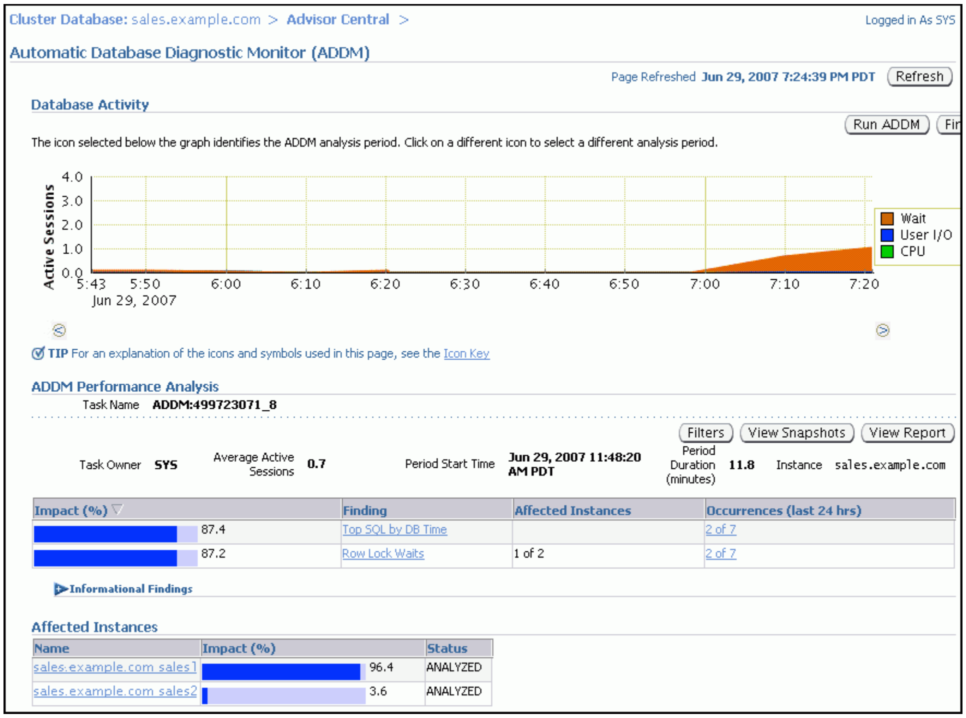


Instances

Name	Status	Host Name	Incidents				ADDM Findings	ASM Instance					
			⊖	⊗	⚠	🚩		Name	Status	⊖	⊗	⚠	🚩
qos1_qos1_1	⇧	node06.example.com	0	0	0	0	0	+ASM3_node06.example.com	⇧	0	0	0	0
qos1_qos1_3	⇧	node05.example.com	0	0	0	0	0	+ASM1_node05.example.com	⇧	0	1	0	0

   When you select the number of ADDM Findings, the Automatic Database Diagnostic Monitor (ADDM) page for the cluster database appears.

3. Review the results of the ADDM run.



Cluster Database: sales.example.com > Advisor Central > Logged in As SYS

Automatic Database Diagnostic Monitor (ADDM)

Page Refreshed Jun 29, 2007 7:24:39 PM PDT (Refresh)

Database Activity

(Run ADDM) (Fin

The icon selected below the graph identifies the ADDM analysis period. Click on a different icon to select a different analysis period.

TIP For an explanation of the icons and symbols used in this page, see the Icon Key

ADDM Performance Analysis

Task Name ADDM:499723071_8

(Filters) (View Snapshots) (View Report)

Task Owner SYS　Average Active Sessions 0.7　Period Start Time Jun 29, 2007 11:48:20 AM PDT　Period Duration (minutes) 11.8　Instance sales.example.com

Impact (%) ▽		Finding	Affected Instances	Occurrences (last 24 hrs)
	87.4	Top SQL by DB Time		2 of 7
	87.2	Row Lock Waits	1 of 2	2 of 7

▷Informational Findings

Affected Instances

Name	Impact (%)		Status
sales.example.com sales1		96.4	ANALYZED
sales.example.com sales2		3.6	ANALYZED

On the Automatic Database Diagnostic Monitor (ADDM) page, the Database Activity chart shows the database activity during the ADDM analysis period. Database activity types are defined in the legend based on its corresponding color in the chart. Each icon below the chart represents a different ADDM task, which in turn corresponds to a pair of individual Oracle Database snapshots saved in the Workload Repository.

In the ADDM Performance Analysis section, the ADDM findings are listed in descending order, from highest impact to least impact. The Informational Findings section lists the areas that do not have a performance impact and are for informational purpose only.

The Affected Instances chart shows how much each instance is impacted by these findings.

4. (Optional) Click the Zoom icons to shorten or lengthen the analysis period displayed on the chart.

5. (Optional) To view the ADDM findings in a report, click **View Report**.

   The View Report page appears.

   You can click **Save to File** to save the report for later access.

6. On the ADDM page, in the Affected Instances table, click the link for the instance associated with the ADDM finding that has the largest value for Impact.

   The Automatic Database Diagnostic Monitor (ADDM) page for that instance appears.

7. In the ADDM Performance Analysis section, select the name of a finding.

   The Performance Findings Detail page appears.

8. View the available Recommendations for resolving the performance problem. Run the SQL Tuning Advisor to tune the SQL statements that are causing the performance findings.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring user activity
- *Oracle Database 2 Day DBA* for more information about tuning a database and instance

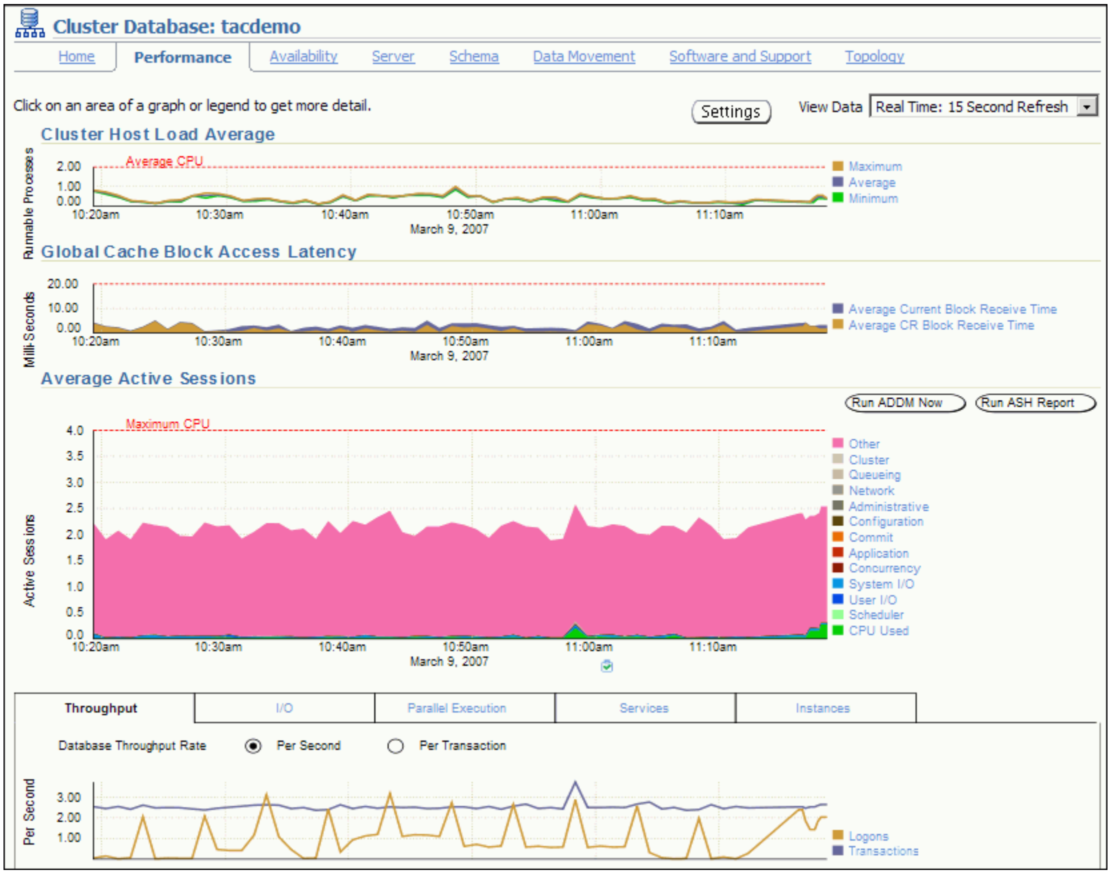
Using the Cluster Database Performance Page

The Cluster Database Performance page of Oracle Enterprise Manager provides a quick glimpse of the performance statistics for a database. Enterprise Manager accumulates data from each instance over specified periods of time, called collection-based data. Enterprise Manager also provides current data from each instance, known as real-time data.

Statistics are rolled up across all the instances in the cluster database. Using the links next to the charts, you can get more specific information and perform any of the following tasks:

- Identify the causes of performance issues.
- Decide whether resources need to be added or redistributed.
- Tune your SQL plan and schema for better optimization.
- Resolve performance issues.

The following screenshot shows a partial view of the Cluster Database Performance page. You access this page by selecting **Performance Home** from **Performance** menu.



Click on an area of a graph or legend to get more detail.

View Data Real Time: 15 Second Refresh

Database Throughput Rate ● Per Second ○ Per Transaction

The following sections describe the charts on the Performance page:

- About the Global Cache Block Access Latency Chart
- Viewing the Chart for Global Cache Block Access Latency
- Viewing the Chart for Cluster Host Load Average
- Viewing the Chart for Average Active Sessions
- Viewing the Database Throughput Chart
- Viewing the Services Chart
- Viewing the Active Sessions by Instance Chart

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database 2 Day + Performance Tuning Guide*

About the Global Cache Block Access Latency Chart

Each cluster database instance has its own buffer cache in its System Global Area (SGA). Using Cache Fusion, Oracle RAC environments logically combine each instance's buffer cache to enable the database instances to process data as if the data resided on a logically combined, single cache.

When a process attempts to access a data block, it first tries to locate a copy of the data block in the local buffer cache. If a copy of the data block is not found in the local buffer cache, then a global cache operation is initiated. Before reading a data block from disk, the process attempts to find the data block in the buffer cache of another instance. If the data block is in the buffer cache of another instance, then Cache Fusion transfers a version of the data block to the local buffer cache, rather than having one database instance write the data block to disk and requiring the other instance to reread the data block from disk. For example, after the `orcl1` instance loads a data block into its buffer cache, the `orcl2` instance can more quickly acquire the data block from the `orcl1` instance by using Cache Fusion rather than by reading the data block from disk.

The Global Cache Block Access Latency chart shows data for two different types of data block requests: current and consistent-read (CR) blocks. When you update data in the database, Oracle Database must locate the most recent version of the data block that contains the data, which is called the **current block**. If you perform a query, then only data committed before the query began is visible to the query. Data blocks that were changed after the start of the query are reconstructed from data in the undo segments. The reconstructed data is made available to the query in the form of a **consistent-read block**.

The Global Cache Block Access Latency chart on the Cluster Database Performance page shows the **latency** for each type of data block request, that is the elapsed time it takes to locate and transfer consistent-read and current blocks between the buffer caches.

If the Global Cache Block Access Latency chart shows high latencies (high elapsed times), then this can be caused by any of the following:

- A high number of requests caused by SQL statements that are not tuned.
- Several processes in the queue waiting for the CPU, or scheduling delays.
- Slow, busy, or faulty interconnects. In these cases, check your network connection for dropped packets, retransmittals, or cyclic redundancy check (CRC) errors.

Concurrent read and write activity on shared data in a cluster is a frequently occurring activity. Depending on the service requirements, this activity does not usually cause performance problems. However, when global cache requests cause a performance problem, optimizing SQL plans and the schema to improve the rate at which data blocks are located in the local buffer cache, and minimizing I/O is a successful strategy for performance tuning. If the latency for consistent-read and current block requests reaches 10 milliseconds, then your first step in resolving the problem should be to go to the Cluster Cache Coherency page for more detailed information.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "Accessing the Cluster Cache Coherency Page"
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*

Viewing the Chart for Global Cache Block Access Latency

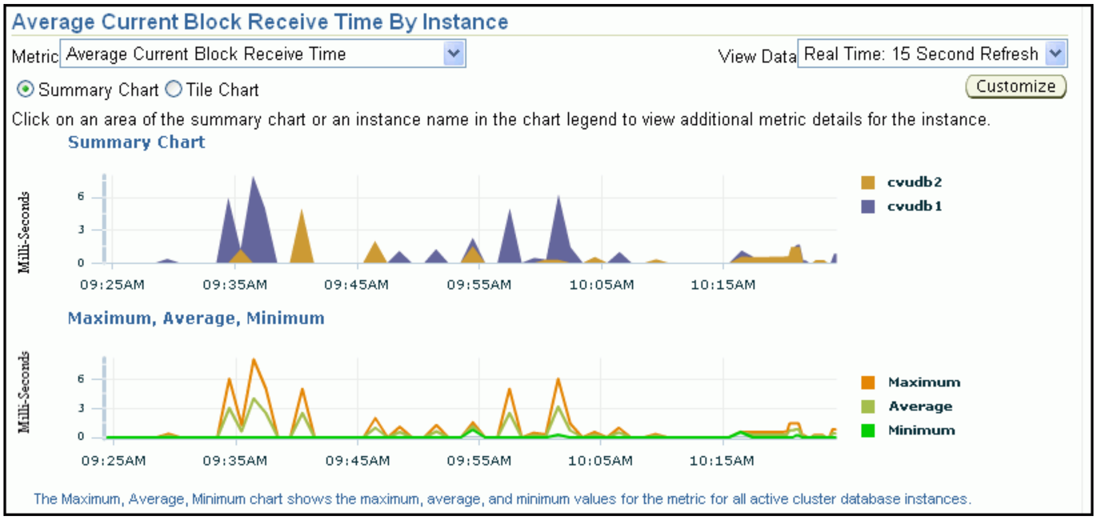
In Oracle Enterprise Manager, on the Global Cache Block Access Latency chart on the Cluster Database Performance page, you can click either metric for the Global Cache Block Access Latency chart to view more detailed information about that type of cached block. For example, if you click the metric **Average Current Block Receive Time**, then the Average Current Block Receive Time by Instance page appears, displaying a summary chart that depicts the average current block receive time for up to four nodes in the cluster. You can select whether the data is displayed in a summary chart or using tile charts. If you choose Summary chart, then, by default, the instances with the four highest receive times are displayed. If you choose Tile charts, then the data for each node is displayed in its own chart. You can customize which nodes are displayed in either the Summary or Tile chart display.

Also, on the Average Current Block Receive Time By Instance page or the Cluster Cache Coherency page, you can use the slider bar on the Active Session History chart to focus on a five minute window (time period) within the past one hour. Using the slider bar enables you to identify the top sessions, services, modules, actions, or SQL statements that were running during a period of high cache coherency activity.

At the top of the page, you can use the Metric list to change the metric displayed. The choices are:

- Average CR Block Receive Time
- Average Current Block Receive Time
- GC Current Blocks Received
- GC CR Blocks Received
- Physical Reads
- Global Cache Block Transfers

Each metric displays a monitoring page for that metric. On each metric monitoring page you can view the data for that metric in either a summary chart or using tile charts. You can also view the Maximum, Average, Minimum chart on the metric monitoring page to view the maximum, average, and minimum values for the metric for all active cluster database instances.



Average Current Block Receive Time By Instance

Metric: Average Current Block Receive Time

View Data: Real Time: 15 Second Refresh

● Summary Chart ○ Tile Chart

Click on an area of the summary chart or an instance name in the chart legend to view additional metric details for the instance.

The Maximum, Average, Minimum chart shows the maximum, average, and minimum values for the metric for all active cluster database instances.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "Accessing the Cluster Cache Coherency Page"
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Real Application Clusters Administration and Deployment Guide*

Viewing the Chart for Cluster Host Load Average

Using Oracle Enterprise Manager, the Cluster Host Load Average chart in the Cluster Database Performance page shows potential problems that are outside the database. The chart shows maximum, average, and minimum load values for available nodes in the cluster for the previous hour.

If the load average is higher than the average of the total number of CPUs across all the hosts in the cluster, then too many processes are waiting for CPU resources. SQL statements that are not tuned often cause high CPU usage. Compare the load average values with the values displayed for CPU Used in the Average Active Sessions chart. If the sessions value is low and the load average value is high, then this indicates that something else on the host, other than your database, is consuming the CPU.

You can click any of the load value labels for the Cluster Host Load Average chart to view more detailed information about that load value. For example, if you click the label **Average**, then the Hosts: Average Load page appears, displaying charts that depict the average host load for up to four nodes in the cluster.

You can select whether the data is displayed in a summary chart, combining the data for each node in one display, or using tile charts, where the data for each node is displayed in its own chart. You can click **Customize** to change the number of tile charts displayed in each row or the method of ordering the tile charts.

For more information about changing the data displayed on the Hosts: Average Load page, refer to the Enterprise Manager online Help.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database 2 Day + Performance Tuning Guide*

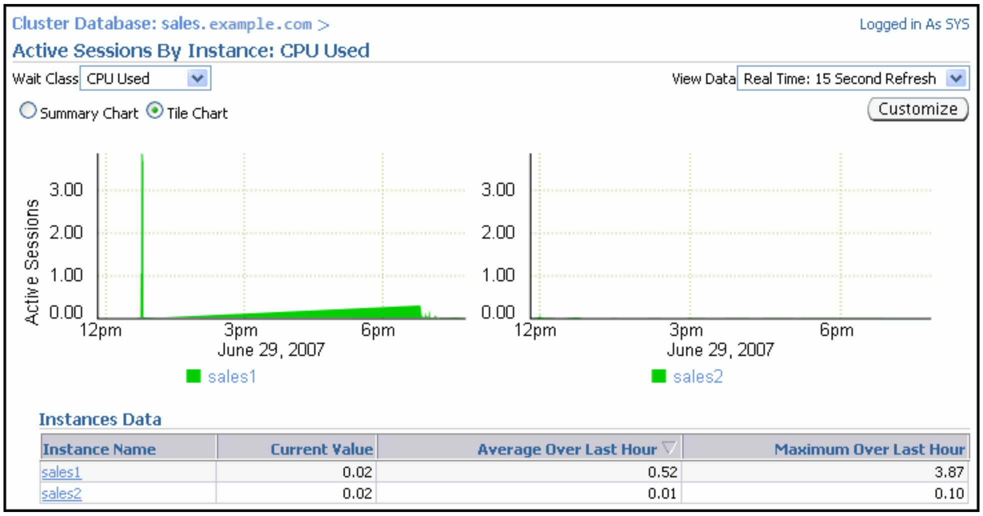
Viewing the Chart for Average Active Sessions

Using Oracle Enterprise Manager, the Average Active Sessions chart in the Cluster Database Performance page shows potential problems inside the database. Categories, called wait classes, show how much of the database is using a resource, such as CPU or disk I/O. Comparing CPU time to wait time helps to determine how much of the response time is consumed with useful work rather than waiting for resources that are potentially held by other processes.

The chart displays the workload on the database or instance and identifies performance issues. At the cluster database level, this chart shows the aggregate wait class statistics across all the instances. For a more detailed analysis, you can click the clipboard icon at the bottom of the chart to view the ADDM analysis for the database for that time period.

Compare the peaks on the Average Active Sessions chart with those on the Database Throughput charts. If the Average Active Sessions chart displays many sessions waiting, indicating internal contention, but throughput is high, then the situation may be acceptable. The database is probably also performing efficiently if internal contention is low but throughput is high. However, if internal contention is high and throughput is low, then consider tuning the database.

If you click the wait class legends beside the Average Active Sessions chart, then you can view instance-level information stored in Active Sessions by Instance pages. You can use the Wait Class action list on the Active Sessions by Instance page to view the different wait classes. The Active Sessions by Instance pages show the service times for up to four instances. Using the Customize button you can select the instances that are displayed. You can view the data for the instances separately using tile charts, or you can combine the data into a single summary chart.



Cluster Database: sales.example.com >

Logged in As SYS

Active Sessions By Instance: CPU Used

Wait Class: CPU Used

View Data: Real Time: 15 Second Refresh

○ Summary Chart ● Tile Chart

Instances Data

Instance Name	Current Value	Average Over Last Hour	Maximum Over Last Hour
sales1	0.02	0.52	3.87
sales2	0.02	0.01	0.10

If you must diagnose and fix problems that are causing the higher number of wait events in a specific category, then you can select an instance of interest and view the wait events, also the SQL, sessions, services, modules, and actions that are consuming the most database resources.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring user activity
- *Oracle Database 2 Day DBA* for more information about tuning a database and instance

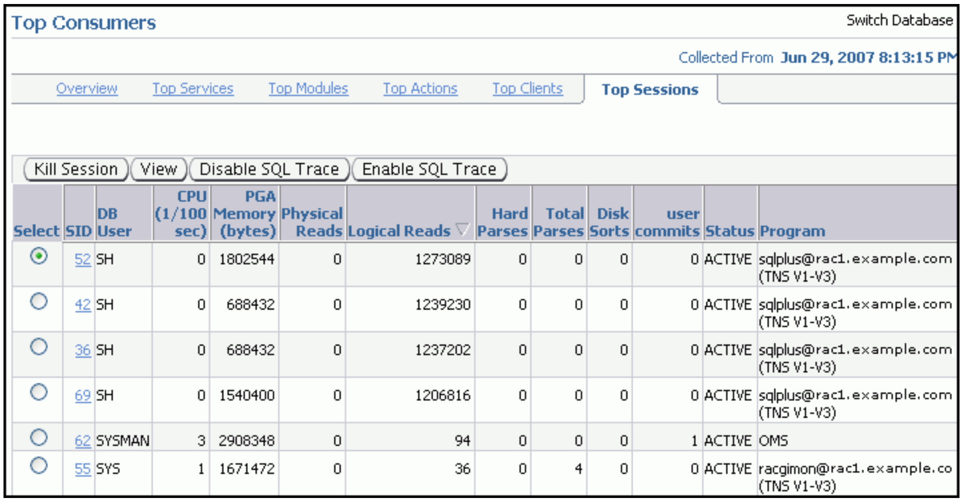
Viewing the Database Throughput Chart

Using Oracle Enterprise Manager, the last chart on the Cluster Database Performance page monitors the usage of various database resources. By clicking the Throughput tab at the top of this chart you can view the Database Throughput chart.

The Database Throughput charts summarize any resource contention that appears in the Average Active Sessions chart, and also show how much work the database is performing on behalf of the users or applications. The Per Second view shows the number of transactions compared to the number of logons, and the amount of physical reads compared to the redo size per second. The Per Transaction view shows the amount of physical reads compared to the redo size per transaction. Logons is the number of users that are logged on to the database.

You can also obtain information at the instance level by clicking a legend to the right of the charts to access the Database Throughput by Instance page. This page shows the breakdown of the aggregated Database Throughput chart for up to four instances. Using the Customize button you can select the instances that are displayed. You can view the data for the instances separately using tile charts, or you can combine the data into a single summary chart. You can use this page to view the throughput for a particular instance, which may help you diagnose throughput problems.

You can drill down further on the Database Throughput by Instance page to see the sessions of an instance consuming the greatest resources. Click an instance name legend just under the chart to go to the Top Sessions subpage of the Top Consumers page for that instance.



Top Consumers

Switch Database

Collected From Jun 29, 2007 8:13:15 PM

Select	SID	DB User	CPU (1/100 sec)	PGA Memory (bytes)	Physical Reads	Logical Reads ▽	Hard Parses	Total Parses	Disk Sorts	user commits	Status	Program
●	52	SH	0	1802544	0	1273089	0	0	0	0	ACTIVE	sqlplus@rac1.example.com (TNS V1-V3)
○	42	SH	0	688432	0	1239230	0	0	0	0	ACTIVE	sqlplus@rac1.example.com (TNS V1-V3)
○	36	SH	0	688432	0	1237202	0	0	0	0	ACTIVE	sqlplus@rac1.example.com (TNS V1-V3)
○	69	SH	0	1540400	0	1206816	0	0	0	0	ACTIVE	sqlplus@rac1.example.com (TNS V1-V3)
○	62	SYSMAN	3	2908348	0	94	0	0	0	1	ACTIVE	OMS
○	55	SYS	1	1671472	0	36	0	4	0	0	ACTIVE	racgimon@rac1.example.co (TNS V1-V3)

For more information about the information on this page, refer to the Enterprise Manager Help system.

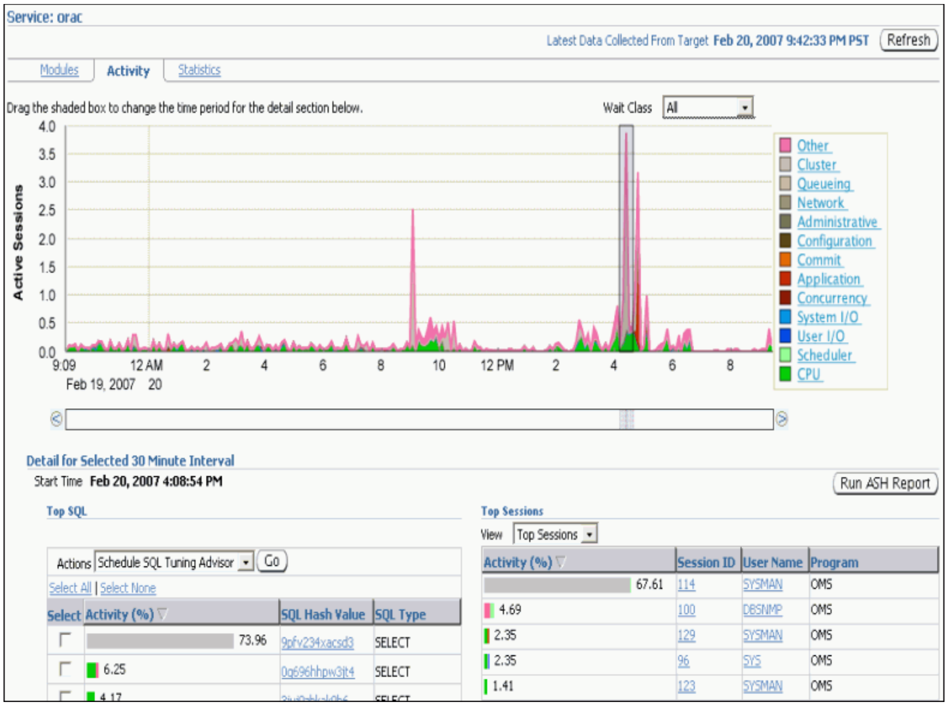
See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring instance activity
- *Oracle Database 2 Day DBA* for more information about tuning a database and instance

Viewing the Services Chart

On the Performance page of Oracle Enterprise Manager, the usage metrics of various database resources are shown in charts. If you click the Services tab at the top of this chart, then you can view the Services chart.

The Services chart shows the top services being used by the active sessions. Only active services are shown. You can select a service legend to the right of the chart to go to the Service subpage of the Top Consumers page. The Activity subtab is selected by default. On this page you can view real-time data showing the session loads by wait classes for the service.



Service: orac

Latest Data Collected From Target Feb 20, 2007 9:42:33 PM PST

Detail for Selected 30 Minute Interval

Start Time Feb 20, 2007 4:08:54 PM

Top SQL

Select	Activity (%)	SQL Hash Value	SQL Type
□	73.96	9pfv234xacsd3	SELECT
□	6.25	0g696hhpw3jt4	SELECT
□	4.17	[illegible]	SELECT

Top Sessions

Activity (%)	Session ID	User Name	Program
67.61	114	SYSMAN	OMS
4.69	100	DBSNMP	OMS
2.35	129	SYSMAN	OMS
2.35	96	SYS	OMS
1.41	123	SYSMAN	OMS

For more information about the information on this page, refer to the Enterprise Manager Help system.

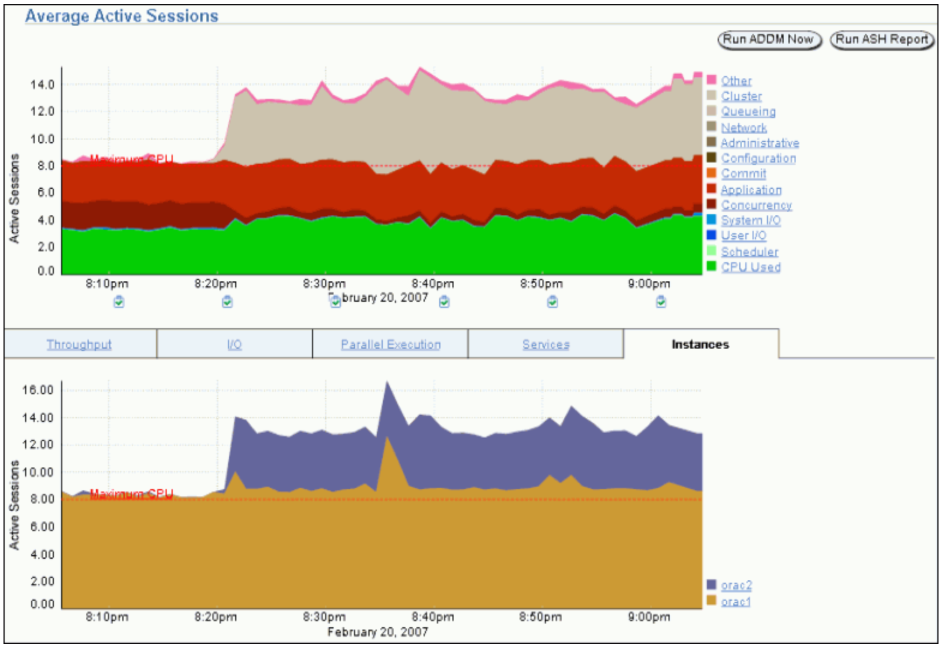
See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database 2 Day DBA* for more information about tuning a database and instance

Viewing the Active Sessions by Instance Chart

On the Performance page of Oracle Enterprise Manager, there is a chart that monitors the usage of various database resources. By clicking the Instances tab at the top of this chart you can view the Active Sessions by Instance chart.

The Active Sessions by Instance chart summarize any resource contention that appears in the Average Active Sessions chart. Using this chart you can quickly determine how much of the database work is being performed on each instance.



You can also obtain information at the instance level by clicking a legend to the right of the chart to access the Top Sessions page. On the Top Session page you can view real-time data showing the sessions that consume the greatest system resources.

For more information about the information on this page, refer to the Enterprise Manager Help system.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database 2 Day DBA* for more information about tuning a database and instance

Viewing Other Performance Related Charts

In the Additional Monitoring Links and Additional Instance Monitoring Links section of the Cluster Database Performance page, there are links to other charts that are useful in evaluating the performance of your cluster database.

This section contains the following topics:

- Accessing the Cluster Cache Coherency Page
- Accessing the Top Consumers Page
- Accessing the Top Sessions Page
- Accessing the Top Activity Page
- Accessing the Instance Activity Page
- Accessing the Top Segments Page
- Accessing the Database Locks Page

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring user activity
- *Oracle Database 2 Day DBA* for more information about tuning a database and instance

Accessing the Cluster Cache Coherency Page

The Cluster Cache Coherency page contains summary charts for cache coherency metrics for the cluster.

The following table describes the Cluster Cache Coherency charts and the actions to perform to access more comprehensive information for problem resolution.

Table 8–1 Cluster Cache Coherency Charts

Name	Description
Global Cache Block Access Latency	Shows the total elapsed time, or latency, for a block request. Click a legend to the right of the chart to view the average time it takes to receive data blocks for each block type (current or CR) by instance. On the Average Block Receive Time by Instance page, you can click an instance legend under the chart to go to the Block Transfer for Local Instance page, where you can identify which block classes, such as undo blocks, data blocks, and so on, are subject to intense global cache activity. This page displays the block classes that are being transferred, and which instances are transferring most of the blocks. Cache transfer indicates how many current and CR blocks for each block class were received from remote instances, including how many transfers incurred a delay (busy) or an unexpected longer delay (congested).
Global Cache Block Transfer Rate	Shows the total aggregated number of blocks received by all instances in the cluster by way of an interconnect. Click a legend to the right of the chart to go to the Global Cache Blocks Received by Instance page for that type of block. From there, you can click an instance legend under the chart to go to the Segment Statistics by Instance page, where you can see which segments are causing cache contention.
Global Cache Block Transfers and Physical Reads	Shows the percentage of logical read operations that retrieved data from the buffer cache of other instances by way of Direct Memory Access and from disk. It is essentially a profile of how much work is performed in the local buffer cache, rather than the portion of remote references and physical reads, which both have higher latencies. Click a legend to the right of the chart to go to the Global Cache Block Transfers vs. Logical Reads by Instance and Physical Reads vs. Logical Reads by Instance pages. From there, you can click an instance legend under the chart to go to the Segment Statistics by Instance page, where you can see which segments are causing cache contention.

To access the Cluster Cache Coherency page:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

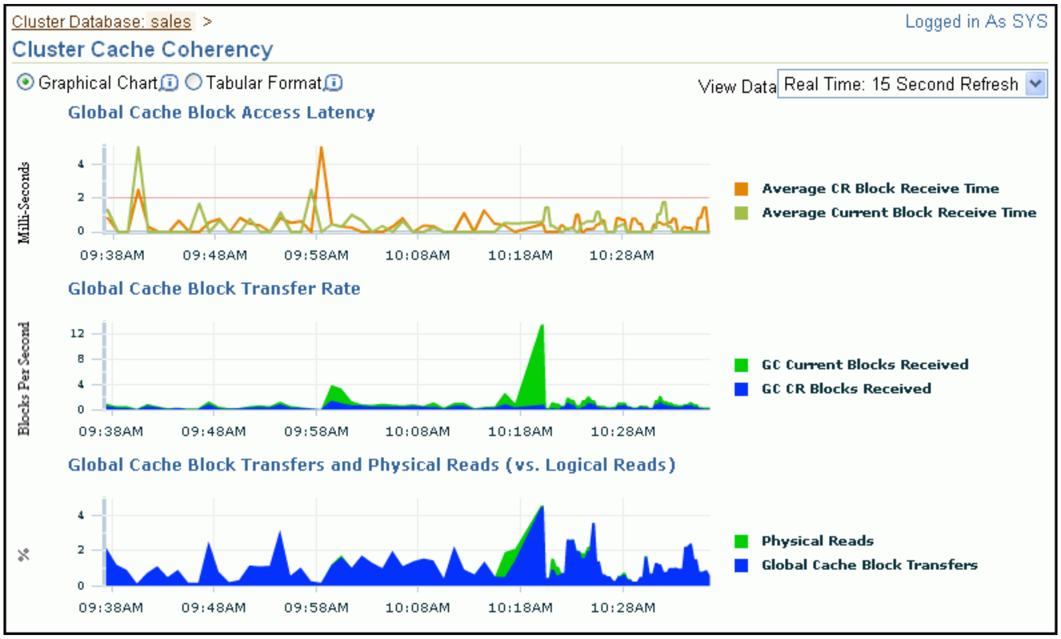
2. From the **Performance** menu, select **Performance Home**.

   The Cluster Database Performance Home page appears.

3. Click **Cluster Cache Coherency** in the Additional Monitoring Links section at the bottom of the page.

4. Alternatively, click either of the legends to the right of the Global Cache Block Access Latency chart.

   The Cluster Cache Coherency page appears.



Cluster Database: sales >

Logged in As SYS

Cluster Cache Coherency

● Graphical Chart ○ Tabular Format

View Data Real Time: 15 Second Refresh

Global Cache Block Access Latency
Milli-Seconds
4
2
0
09:38AM
09:48AM
09:58AM
10:08AM
10:18AM
10:28AM
🟧 Average CR Block Receive Time
🟩 Average Current Block Receive Time
Global Cache Block Transfer Rate
Blocks Per Second
12
8
4
0
09:38AM
09:48AM
09:58AM
10:08AM
10:18AM
10:28AM
🟩 GC Current Blocks Received
🟦 GC CR Blocks Received
Global Cache Block Transfers and Physical Reads (vs. Logical Reads)
%
4
2
0
09:38AM
09:48AM
09:58AM
10:08AM
10:18AM
10:28AM
🟩 Physical Reads
🟦 Global Cache Block Transfers

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring instance activity

Accessing the Top Consumers Page

The Top Consumers page provides access to several tabs that enable you to view real-time or collection-based data for the services, modules, actions, clients, and sessions that are consuming the most system resources.

To access the Top Consumers page:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

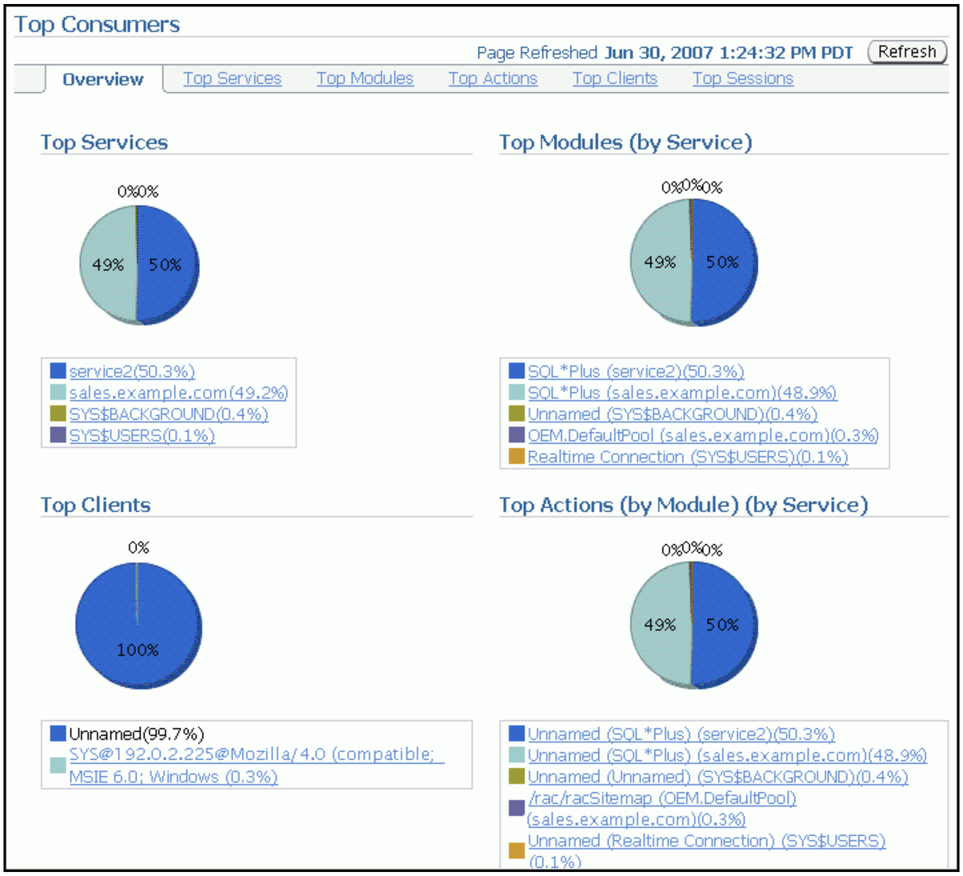
   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. From the **Performance** menu, select **Performance** Home.

   The Cluster Database Performance Home page appears.

3. Click **Top Consumers** in the Additional Monitoring Links section at the bottom of the page.

When accessed this way, the Top Consumers page initially displays the Overview tab by default, which shows aggregated summary data for the highest resource consumers.


Top Consumers
Page Refreshed Jun 30, 2007 1:24:32 PM PDT
Refresh
Overview
Top Services
Top Modules
Top Actions
Top Clients
Top Sessions
Top Services
0%0%
49%
50%
service2(50.3%)
sales.example.com(49.2%)
SYS$BACKGROUND(0.4%)
SYS$USERS(0.1%)
Top Modules (by Service)
0%0%0%
49%
50%
SQL*Plus (service2)(50.3%)
SQL*Plus (sales.example.com)(48.9%)
Unnamed (SYS$BACKGROUND)(0.4%)
OEM.DefaultPool (sales.example.com)(0.3%)
Realtime Connection (SYS$USERS)(0.1%)
Top Clients
0%
100%
Unnamed(99.7%)
SYS@192.0.2.225@Mozilla/4.0 (compatible; MSIE 6.0; Windows (0.3%)
Top Actions (by Module) (by Service)
0%0%0%
49%
50%
Unnamed (SQL*Plus) (service2)(50.3%)
Unnamed (SQL*Plus) (sales.example.com)(48.9%)
Unnamed (Unnamed) (SYS$BACKGROUND)(0.4%)
/rac/racSitemap (OEM.DefaultPool) (sales.example.com)(0.3%)
Unnamed (Realtime Connection) (SYS$USERS) (0.1%)

4. (Optional) Click the portion of a chart representing the consumer or click the link under the chart for that consumer to view instance-level information about that consumer.

   The page that appears shows the running instances that are serving the consumer.

5. (Optional) Expand the names in the Action or Module column to show data for individual instances.

> **See Also:**
>
> - "Monitoring Oracle RAC Database and Cluster Performance"
> - *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring real-time database performance

Accessing the Top Sessions Page

The Top Sessions page shows a real-time summary list of sessions based on aggregated data. You can see which sessions have consumed the greatest amount of system resources, referred to as the top sessions, and then decide whether you want to stop the sessions.

To access the Top Sessions page:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. From the **Performance** menu, select **Performance** Home.

   The Cluster Database Performance Home page appears.

3. Click **Top Consumers** in the Additional Monitoring Links section at the bottom of the page.

4. On the Top Consumers page, click the **Top Sessions** subtab.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring the top database sessions

Accessing the Top Activity Page

The Top Activity page enables you to view the cluster database activity by waits, services and instances. Also, you can see the details for the Top SQL and Top Sessions for a specific five minute interval by moving the slider bar on the Top Activity chart.

In the Top SQL detail section, you can select problematic SQL statements and either schedule the SQL Tuning Advisor for those statements or create a SQL Tuning Set.

By default, the Top Sessions for the selected time period are shown. Using the View action list in this section you can change the display to one of the following:

- Top Sessions
- Top Services
- Top Modules
- Top Actions
- Top Clients
- Top Files
- Top Objects
- Top PL/SQL
- Top Instances

To access the Top Activity page:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. From the **Performance** menu, select **Performance** Home.

   The Cluster Database Performance Home page appears.

3. Click **Top Activity** in the Additional Monitoring Links section at the bottom of the page.

   The Top Activity page appears.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring user activity
- *Oracle Database 2 Day DBA* for more information about tuning a database and instance

Accessing the Instance Activity Page

The Instance Activity page enables you to view instance activity for several metrics within general metric categories, such as cursors, transactions, sessions, logical I/O, physical I/O, and net I/O. You can view data for each second or transaction.

To access the Instance Activity page:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

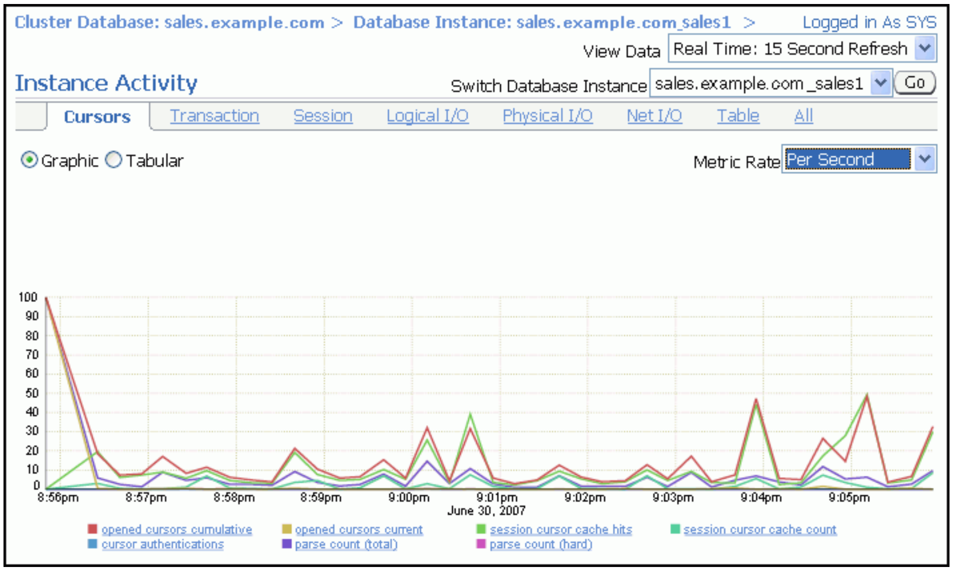
   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. From the **Performance** menu, select **Performance** Home.

   The Cluster Database Performance Home page appears.

3. Click **Instance Activity** in the Additional Monitoring Links section at the bottom of the page.

4. (Optional) Click a metric legend under the chart if in Graphic mode, or click a name in the summary table if in Tabular mode to access top sessions statistics for a particular metric.



Cluster Database: sales.example.com > Database Instance: sales.example.com_sales1 >

Logged in As SYS

View Data: Real Time: 15 Second Refresh

Instance Activity

Switch Database Instance: sales.example.com_sales1

● Graphic ○ Tabular

Metric Rate: Per Second

5. (Optional) Use the **Switch Database Instance** list to change the instance for which the data is displayed in the chart.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring instance activity

Accessing the Top Segments Page

Collecting and viewing statistics at the segment level is an effective method for identifying frequently accessed tables or indexes in a database. The Top Segments page enables you to gather segment-level statistics to identify performance problems associated with individual segments. This page is particularly useful for Oracle RAC, because it also tracks the number of consistent-read and current blocks received by an object. A high number of current blocks received plus a high number of buffer waits may indicate potential resource contention.

To access the Top Segments page:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

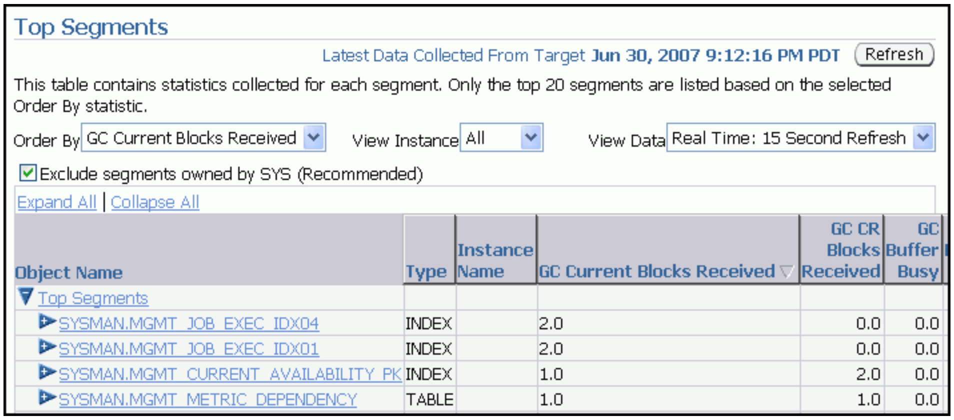
   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. From the **Performance** menu, select **Performance** Home.

   The Cluster Database Performance Home page appears.

3. Click **Top Segments** in the Additional Monitoring Links section at the bottom of the page.

   You can view segments for all instances, or use a filter to see segments for a specific instance.



Top Segments

Latest Data Collected From Target **Jun 30, 2007 9:12:16 PM PDT**

This table contains statistics collected for each segment. Only the top 20 segments are listed based on the selected Order By statistic.

Order By: GC Current Blocks Received

View Instance: All

View Data: Real Time: 15 Second Refresh

☑ Exclude segments owned by SYS (Recommended)

Object Name	Type	Instance Name	GC Current Blocks Received ▽	GC CR Blocks Received	GC Buffer Busy
▼ Top Segments					
▶ SYSMAN.MGMT_JOB_EXEC_IDX04	INDEX		2.0	0.0	0.0
▶ SYSMAN.MGMT_JOB_EXEC_IDX01	INDEX		2.0	0.0	0.0
▶ SYSMAN.MGMT_CURRENT_AVAILABILITY_PK	INDEX		1.0	2.0	0.0
▶ SYSMAN.MGMT_METRIC_DEPENDENCY	TABLE		1.0	1.0	0.0

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"

Accessing the Database Locks Page

Use the Database Locks page to determine if multiple instances are holding locks for the same object. The page shows user locks, all database locks, or locks that are blocking other users or applications. You can use this information to stop a session that is unnecessarily locking an object.

To access the Database Locks page:

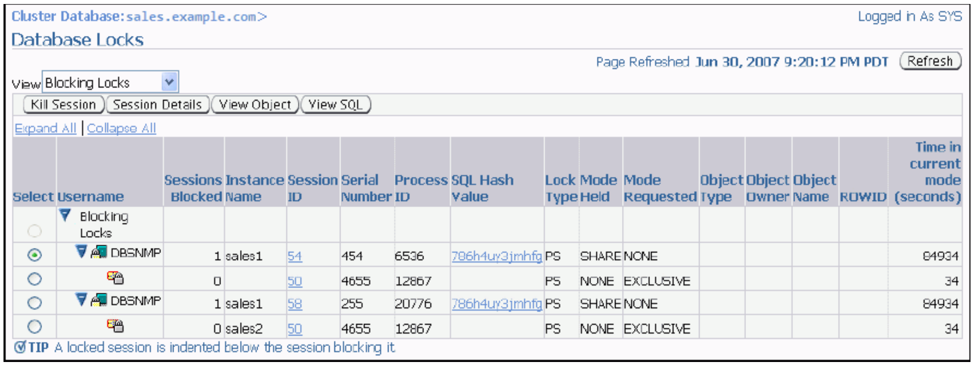
1. Access the Cluster Database Home page of Oracle Enterprise Manager.

   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. From the **Performance** menu, select **Performance** Home.

   The Cluster Database Performance Home page appears.

3. Click **Database Locks** in the Additional Monitoring Links section at the bottom of the page.



Cluster Database: sales.example.com>

Logged in As SYS

Database Locks

Page Refreshed Jun 30, 2007 9:20:12 PM PDT

Select	Username	Sessions Blocked	Instance Name	Session ID	Serial Number	Process ID	SQL Hash Value	Lock Type	Mode Held	Mode Requested	Object Type	Object Owner	Object Name	ROWID	Time in current mode (seconds)
○	Blocking Locks														
●	DBSNMP	1	sales1	54	454	6536	786h4uy3jmhfg	PS	SHARE	NONE					84934
○		0		50	4655	12867		PS	NONE	EXCLUSIVE					34
○	DBSNMP	1	sales1	58	255	20776	786h4uy3jmhfg	PS	SHARE	NONE					84934
○		0	sales2	50	4655	12867		PS	NONE	EXCLUSIVE					34

TIP A locked session is indented below the session blocking it

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- *Oracle Database Administrator's Guide* for more information about monitoring locks

Viewing the Cluster Database Topology

Using Enterprise Manager you can view a graphical representation of your cluster environment. Using the topology view you can quickly see the components that comprise your cluster database environment, such as database instances, listeners, Oracle ASM instances, hosts, and interfaces.

After you click the topology chart to activate the controls, you can mouse over a component to see the status and configuration details for that component. If you select a component in the topology chart, then you can then right-click that component to view a set of menu actions specific for that component.

To view the topology for your cluster database environment:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

   See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. From the **Cluster Database** menu, select **Configuration**, and then **Topology**.

   The Cluster Database Topology page appears.

3. (Optional) Move the mouse cursor over any component in the topology diagram to display information about that component in a popup box.

4. Select any component in the topology diagram to change the information displayed in the Selection Details section.

5. (Optional) Click **Legend** at the bottom of the page, on the left-hand side, to display the Topology Legend page.

   This page describes the icons used in Cluster Topology and Cluster Database Topology.

6. (Optional) Right-click the currently selected component to view the menu actions available for that component.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance" on page 8-1
- "Viewing the Cluster Topology Page" on page 8-30
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC" on page 1-3

Monitoring Oracle Clusterware

You can use Enterprise Manager to monitor Oracle Clusterware. Other monitoring features available with Enterprise Manager include:

- Viewing the status of Oracle Clusterware on each node of the cluster
- Receiving notifications if there are any VIP relocations
- Monitoring the overall throughput across the private interconnect
- Receiving notifications if nodeapps go down or come up
- Viewing alerts if a database instance is using the public interface instead of the VIP
- Monitoring the Clusterware alert log for OCR or voting disk related issues, node evictions, and other clusterware errors

You can also monitor Oracle Clusterware resources and use cluster component health monitoring for individual components. To monitor resources you click the Administration link from the Cluster home page. To monitor the health of individual cluster components you click the All Metrics link in the Related Links section of the Cluster home page.

This section contains the following topics:

- Accessing the Oracle Clusterware Information
- Reviewing the Oracle Clusterware Home Page
- Viewing the Cluster Performance Page
- Viewing the Cluster Targets Page
- Viewing the Cluster Administration Page
- About the Cluster Interconnects Page
- Viewing the Cluster Topology Page

See Also:

- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "About Verifying the Oracle Clusterware Installation"
- "Using CRSCTL to Diagnose Cluster Issues"

Accessing the Oracle Clusterware Information

From the Cluster Database Home page, there a several ways to access Oracle Clusterware information.

To access Oracle Clusterware information:

1. Access the Cluster Database Home page of Oracle Enterprise Manager.

See *Oracle Database 2 Day DBA* for details on logging in to Enterprise Manager.

2. In the General section, click the link next to **Cluster** to view the Cluster Home page.
3. On the Cluster Home page, in the Diagnostic Summary section, click the number next to **Interconnect Alerts** to view the Interconnects subpage for the cluster.

   Alternatively, from the Cluster menu, select **Interconnects** to display the Interconnects page.
4. From the Cluster menu, select **Configuration**, and then select **Topology**. Click a node in the graphical display to activate the controls. Click the **Interface** component. Right-click the Interface component, then choose **View Details** from the menu to display the Interconnects page for the cluster.
5. In the Cluster Databases section of the Cluster Home page, select the name of a cluster database to return to the Cluster Database Home page for that database.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance" on page 8-1
- "Viewing the Cluster Topology Page" on page 8-30
- "About the Cluster Interconnects Page" on page 8-28

Reviewing the Oracle Clusterware Home Page

The Cluster Home page enables you to monitor the health and workload of your cluster. It provides a central place for general cluster state information and is updated periodically.

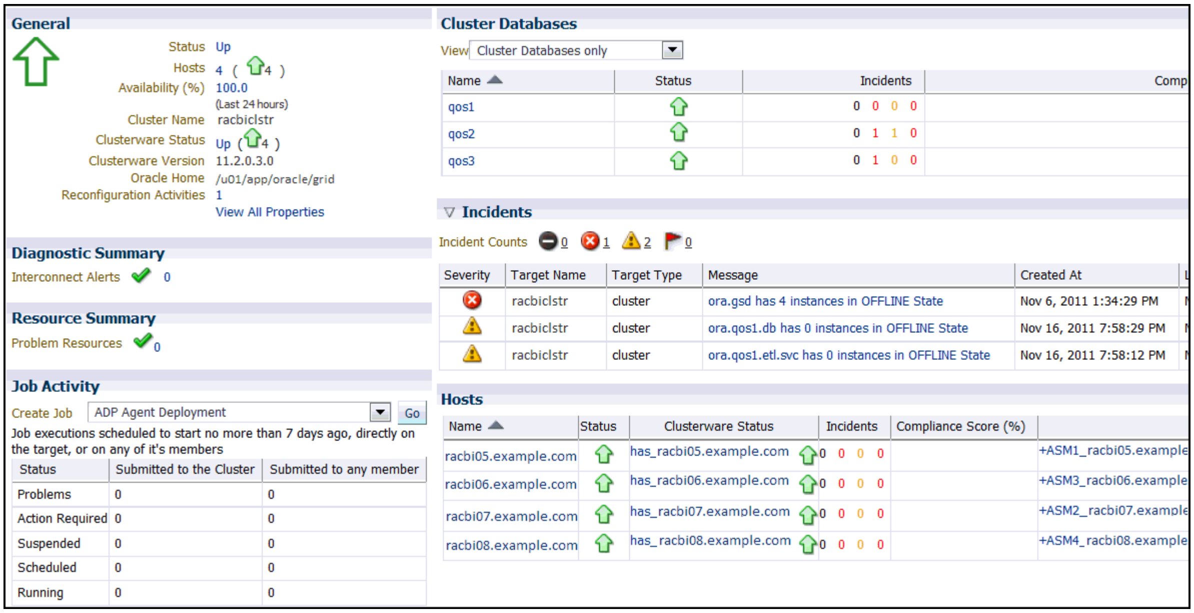
The various sections of the Cluster Home page provide information about the cluster environment and status of the hosts, targets, and clusterware components. For example, the Alerts and Diagnostic Summary sections warn you of errors and performance problems that are impacting the operation of your cluster. You can click the provided links to see more detail about the problem areas.

To monitor the general state of the cluster:

1. From the Cluster Database Home page, in the General section, click the link next to **Cluster**.

   The Cluster Home page appears. An example of the Cluster Home page is shown in Figure 8–1.

Figure 8–1 Cluster Home Page in Oracle Enterprise Manager



General

Status	Up
Hosts	4 (⇧4)
Availability (%)	100.0 (Last 24 hours)
Cluster Name	racbiclstr
Clusterware Status	Up (⇧4)
Clusterware Version	11.2.0.3.0
Oracle Home	/u01/app/oracle/grid
Reconfiguration Activities	1
	View All Properties

Diagnostic Summary

Interconnect Alerts ✓ 0

Resource Summary

Problem Resources ✓ 0

Job Activity

Create Job ADP Agent Deployment ▼ Go

Job executions scheduled to start no more than 7 days ago, directly on the target, or on any of it's members

Status	Submitted to the Cluster	Submitted to any member
Problems	0	0
Action Required	0	0
Suspended	0	0
Scheduled	0	0
Running	0	0

Cluster Databases

View Cluster Databases only ▼

Name ▲	Status	Incidents	Comp
qos1	⇧	0 0 0 0	
qos2	⇧	0 1 1 0	
qos3	⇧	0 1 0 0	

▽ **Incidents**

Incident Counts ⊖ 0 ✖ 1 ⚠ 2 ⚑ 0

Severity	Target Name	Target Type	Message	Created At
✖	racbiclstr	cluster	ora.gsd has 4 instances in OFFLINE State	Nov 6, 2011 1:34:29 PM
⚠	racbiclstr	cluster	ora.qos1.db has 0 instances in OFFLINE State	Nov 16, 2011 7:58:29 PM
⚠	racbiclstr	cluster	ora.qos1.etl.svc has 0 instances in OFFLINE State	Nov 16, 2011 7:58:12 PM

Hosts

Name ▲	Status	Clusterware Status	Incidents	Compliance Score (%)	
racbi05.example.com	⇧	has_racbi05.example.com ⇧	0 0 0 0		+ASM1_racbi05.example
racbi06.example.com	⇧	has_racbi06.example.com ⇧	0 0 0 0		+ASM3_racbi06.example
racbi07.example.com	⇧	has_racbi07.example.com ⇧	0 0 0 0		+ASM2_racbi07.example
racbi08.example.com	⇧	has_racbi08.example.com ⇧	0 0 0 0		+ASM4_racbi08.example

2. (Optional) Click the **Refresh** button to update the information displayed.

   The date and time that data was last collected from the cluster is displayed to the left of the Refresh button.

3. Get a quick view of the cluster in the General section, which includes the following information:

   - Status of the cluster, Up or Down

     Click the **Status** link to drill down to cluster availability details.

   - Number of hosts in the cluster
   - Cluster name
   - The status of Oracle Clusterware overall and the hosts
   - Oracle Clusterware version
   - Oracle Clusterware home directory

4. In the Configuration section, use the **View** list to select which of the following information is displayed for the available hosts in the cluster:

   - Operating Systems (including Hosts and OS Patches)
   - Hardware (including hardware configuration and hosts)

   Click the links under **Host** or **OS Patches** for detailed information.

5. View the Diagnostic Summary section which contains the number of active Interconnect alerts. Click the number of alerts to view the Interconnects subpage.

6. Investigate the Cluster Databases table to view the cluster databases associated with this cluster, their availability, any alerts or policy violations on those databases, their security compliance score, and the database software version.

7. View the Alerts section, which includes the following items:

   - Category list

Optionally choose a category from the list to view only alerts in that category.

- Critical

  This is the number of metrics that have crossed critical thresholds plus the number of other critical alerts, such as those caused by incidents (critical errors).

- Warning

  This is the number of metrics that have crossed warning thresholds.

- Alerts table

  The Alerts table provides information about any alerts that have been issued along with the severity rating of each. Click the alert message in the Message column for more information about the alert.

  When an alert is triggered, the name of the metric for which the alert was triggered is displayed in the Name column. The severity icon for the alert (Warning or Critical) is displayed, along with the time the alert was triggered, the value of the alert, and the time the metric's value was last checked.

8. View the date of the Last Security Evaluation and the Compliance score for the cluster in the Security section.

   The compliance score is a value between 0 and 100 where 100 is a state of complete compliance to the security policies. The compliance score calculation for each target and policy combination to a great extent is influenced by the severity of the violation and importance of the policy, and to a lesser extent by the percentage of violating rows over the total number of rows tested.

9. Review the status of any jobs submitted to the cluster within the last seven days in the Job Activity section.

10. Determine if there are patches to be applied to Oracle Clusterware by reviewing the Critical Patch Advisories for Oracle Homes section.

    To view available patches, you must have first configured your My Oracle Support Credentials as discussed in "Verifying My Oracle Support Credentials" on page 3-3.

11. View basic performance statistics for each host in the cluster in the Hosts table at the bottom of the page.

    Click any link in this table to view further details about that statistic.

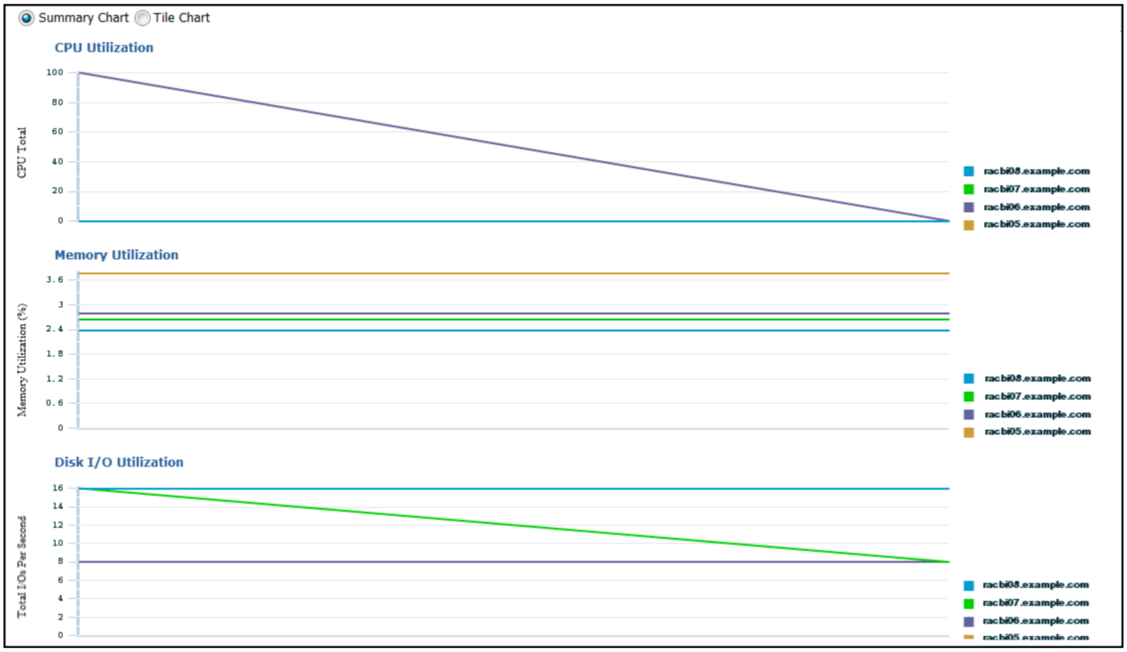
12. Use the **Cluster** menu at the top of the page to view detailed information for Performance, Targets, Interconnects, or Topology.

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance" on page 8-1
- "Viewing the Cluster Topology Page" on page 8-30
- "Obtaining the Patch" on page 10-3
- "Verifying Operating System and Software Requirements" on page 2-7

Viewing the Cluster Performance Page

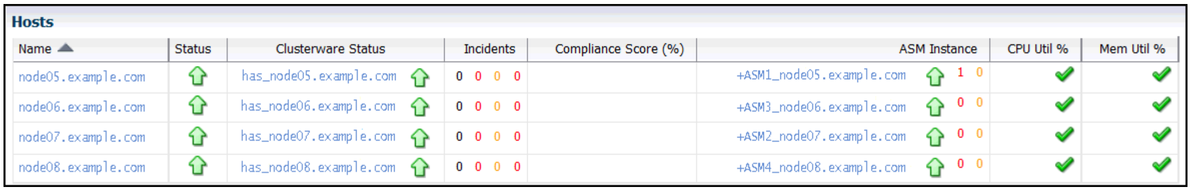
The Cluster Performance page displays utilization statistics, such as CPU, Memory, and Disk I/O, during the past hour for all hosts of a cluster, which is part of the greater Enterprise Manager environment. With this information, you can determine whether you need to add or redistribute resources.



Using the charts on the Cluster Performance page, you can:

- View the CPU, Memory, and Disk I/O charts for the cluster across all hosts.
- View the CPU, Memory, and Disk I/O charts for each host individually by clicking the host name in the legend to the right of the chart.

The Cluster Performance page also contains a Hosts table. The Hosts table displays summary information for the hosts for the cluster, their availability, any alerts on those hosts, CPU and memory utilization percentage, and total I/O per second. You can click a host name in the Hosts table to go to the performance page for that host.



Hosts

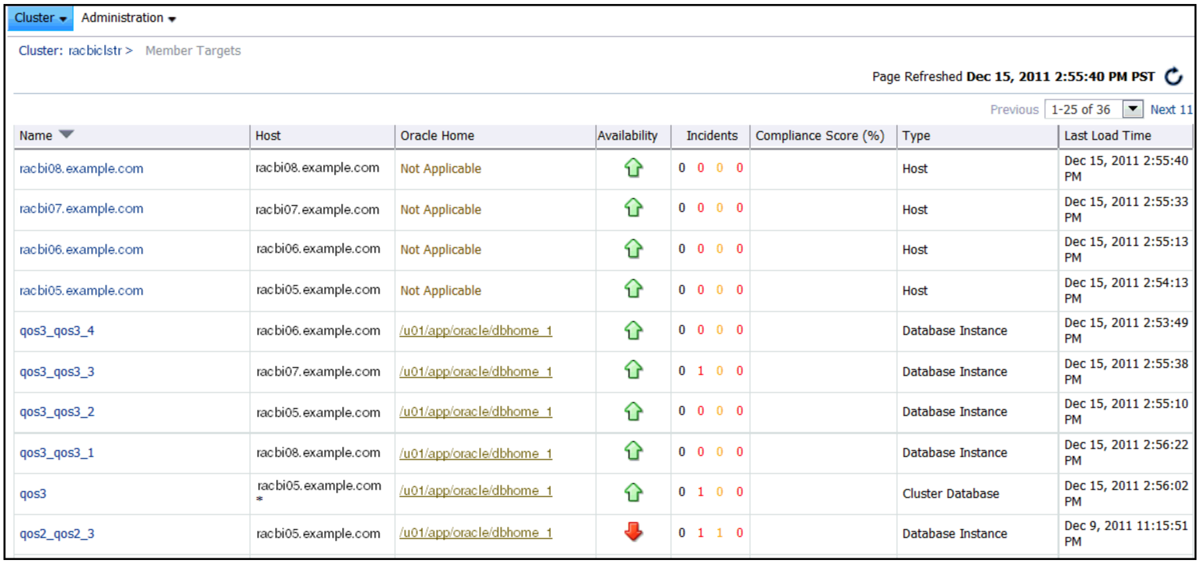
Name ▲	Status	Clusterware Status	Incidents	Compliance Score (%)	ASM Instance	CPU Util %	Mem Util %
node05.example.com	⇧	has_node05.example.com ⇧	0 0 0 0		+ASM1_node05.example.com ⇧ 1 0	✓	✓
node06.example.com	⇧	has_node06.example.com ⇧	0 0 0 0		+ASM3_node06.example.com ⇧ 0 0	✓	✓
node07.example.com	⇧	has_node07.example.com ⇧	0 0 0 0		+ASM2_node07.example.com ⇧ 0 0	✓	✓
node08.example.com	⇧	has_node08.example.com ⇧	0 0 0 0		+ASM4_node08.example.com ⇧ 0 0	✓	✓

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance" on page 8-1
- "Using the Cluster Database Performance Page" on page 8-6

Viewing the Cluster Targets Page

The Cluster Targets page provides a complete list of all targets on the cluster. The table includes the target name, type, host, and location, also the target's availability, warning and critical alerts, and last load time.



Page Refreshed **Dec 15, 2011 2:55:40 PM PST**

Name	Host	Oracle Home	Availability	Incidents	Compliance Score (%)	Type	Last Load Time
racbi08.example.com	racbi08.example.com	Not Applicable	↑	0 0 0 0		Host	Dec 15, 2011 2:55:40 PM
racbi07.example.com	racbi07.example.com	Not Applicable	↑	0 0 0 0		Host	Dec 15, 2011 2:55:33 PM
racbi06.example.com	racbi06.example.com	Not Applicable	↑	0 0 0 0		Host	Dec 15, 2011 2:55:13 PM
racbi05.example.com	racbi05.example.com	Not Applicable	↑	0 0 0 0		Host	Dec 15, 2011 2:54:13 PM
qos3_qos3_4	racbi06.example.com	/u01/app/oracle/dbhome_1	↑	0 0 0 0		Database Instance	Dec 15, 2011 2:53:49 PM
qos3_qos3_3	racbi07.example.com	/u01/app/oracle/dbhome_1	↑	0 1 0 0		Database Instance	Dec 15, 2011 2:55:38 PM
qos3_qos3_2	racbi05.example.com	/u01/app/oracle/dbhome_1	↑	0 0 0 0		Database Instance	Dec 15, 2011 2:55:10 PM
qos3_qos3_1	racbi08.example.com	/u01/app/oracle/dbhome_1	↑	0 0 0 0		Database Instance	Dec 15, 2011 2:56:22 PM
qos3	racbi05.example.com *	/u01/app/oracle/dbhome_1	↑	0 1 0 0		Cluster Database	Dec 15, 2011 2:56:02 PM
qos2_qos2_3	racbi05.example.com	/u01/app/oracle/dbhome_1	↓	0 1 1 0		Database Instance	Dec 9, 2011 11:15:51 PM

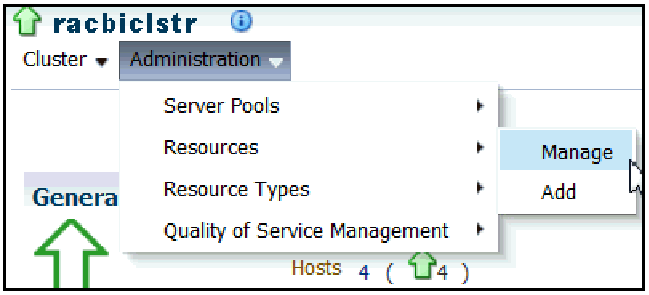
Click a target name to go to the home page for that target. Click the links in the table to get more information about that particular item, alert, or metric.

The Hosts table displays the hosts for the cluster, their availability, any alerts on those hosts, CPU and memory utilization percentage, and total I/O per second.

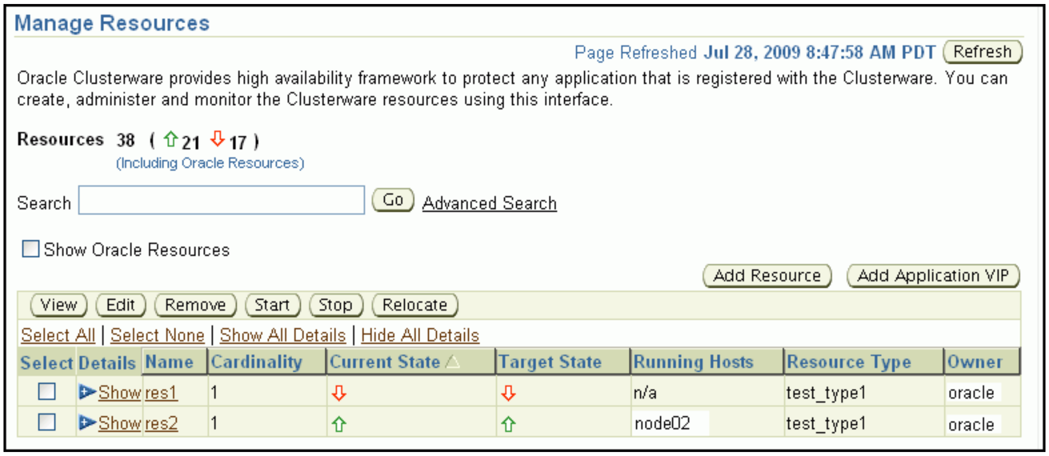
See Also: "Viewing the Cluster Database Topology" on page 8-21

Viewing the Cluster Administration Page

Using the Cluster Administration menu you can manage Oracle Clusterware resources, create resource types, and manage server pools. You can also use this page to configure Oracle Database Quality of Service Management.



If you select Resources from the Administration menu, then the Resources page is displayed. On this page you can start, stop, or relocate the resources that are registered with Oracle Clusterware.



Page Refreshed **Jul 28, 2009 8:47:58 AM PDT**

Oracle Clusterware provides high availability framework to protect any application that is registered with the Clusterware. You can create, administer and monitor the Clusterware resources using this interface.

Resources 38 (⇧21 ⇩17)
(Including Oracle Resources)

Search

□ Show Oracle Resources

Select	Details	Name	Cardinality	Current State △	Target State	Running Hosts	Resource Type	Owner
□	▶Show	res1	1	⇩	⇩	n/a	test_type1	oracle
□	▶Show	res2	1	⇧	⇧	node02	test_type1	oracle

See Also:

- "Viewing the Cluster Database Topology" on page 8-21
- *Oracle Database Quality of Service Management User's Guide* for information about using Oracle Database QoS Management to manage the resources on your system
- *Oracle Clusterware Administration and Deployment Guide* for more information about monitoring Oracle Clusterware

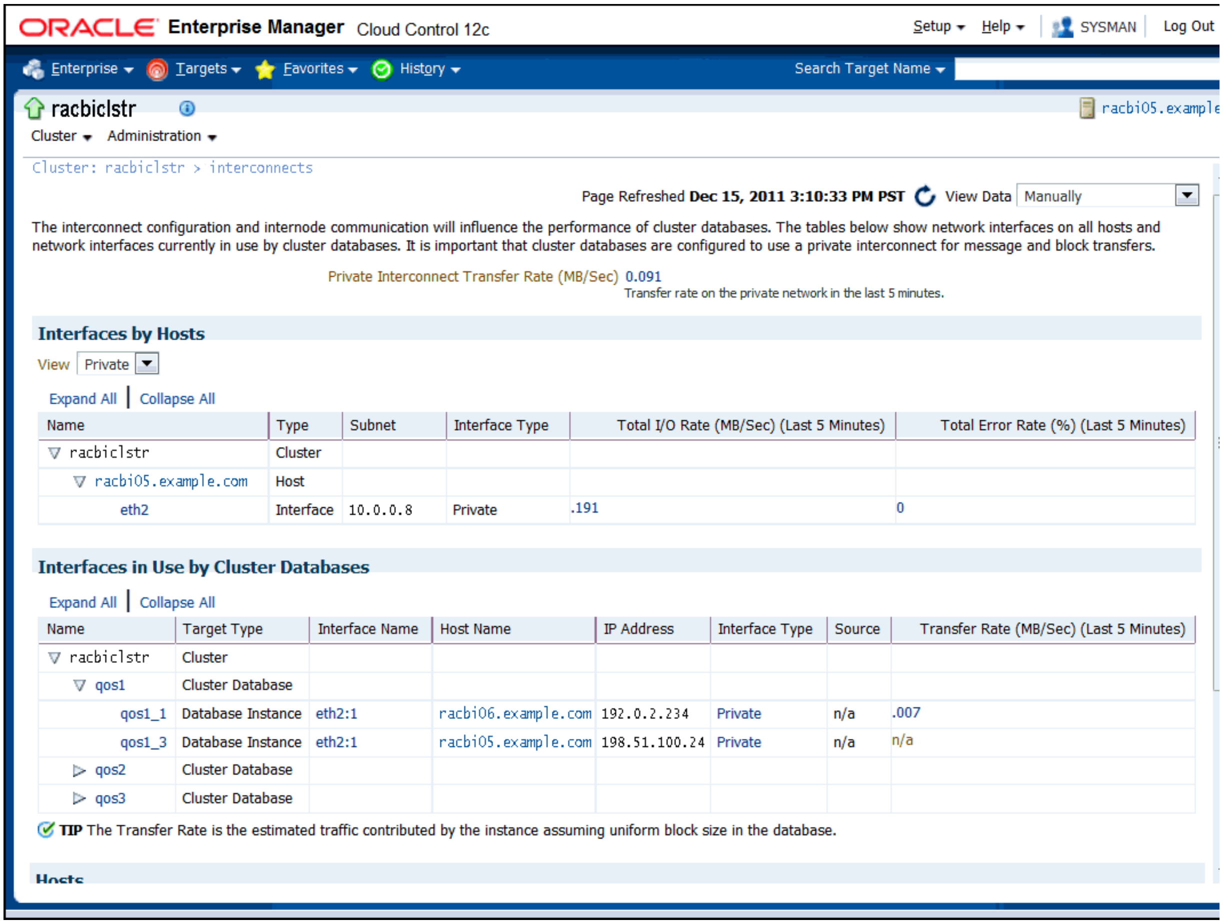
About the Cluster Interconnects Page

The Cluster Interconnects page is useful for monitoring the interconnect interfaces, determining configuration issues, and identifying transfer rate-related issues, including excess traffic. This page helps determine the load added by individual instances and databases on the interconnect. Sometimes you can immediately identify interconnect delays that are due to applications outside Oracle.

You can use this page, shown in Figure 8–2, to perform the following tasks:

- View all interfaces that are configured across the cluster.
- View statistics for the interfaces, such as absolute transfer rates and errors.
- Determine the type of interfaces, such as private or public.
- Determine whether the instance is using a public or private network.
- Determine which database instance is currently using which interface.
- Determine how much the instance is contributing to the transfer rate on the interface.

Figure 8–2 Cluster Interconnects Page in Oracle Enterprise Manager



Page Refreshed **Dec 15, 2011 3:10:33 PM PST** View Data Manually

The interconnect configuration and internode communication will influence the performance of cluster databases. The tables below show network interfaces on all hosts and network interfaces currently in use by cluster databases. It is important that cluster databases are configured to use a private interconnect for message and block transfers.

Private Interconnect Transfer Rate (MB/Sec) 0.091
Transfer rate on the private network in the last 5 minutes.

Interfaces by Hosts

View Private

Expand All | Collapse All

Name	Type	Subnet	Interface Type	Total I/O Rate (MB/Sec) (Last 5 Minutes)	Total Error Rate (%) (Last 5 Minutes)
▽ racbiclstr	Cluster				
▽ racbi05.example.com	Host				
eth2	Interface	10.0.0.8	Private	.191	0

Interfaces in Use by Cluster Databases

Expand All | Collapse All

Name	Target Type	Interface Name	Host Name	IP Address	Interface Type	Source	Transfer Rate (MB/Sec) (Last 5 Minutes)
▽ racbiclstr	Cluster						
▽ qos1	Cluster Database						
qos1_1	Database Instance	eth2:1	racbi06.example.com	192.0.2.234	Private	n/a	.007
qos1_3	Database Instance	eth2:1	racbi05.example.com	198.51.100.24	Private	n/a	n/a
▷ qos2	Cluster Database						
▷ qos3	Cluster Database						

✓ **TIP** The Transfer Rate is the estimated traffic contributed by the instance assuming uniform block size in the database.

Hosts

The Private Interconnect Transfer Rate value shows a global view of the private interconnect traffic, which is the estimated traffic on all the private networks in the cluster. The traffic is calculated as the summary of the input rate of all private interfaces known to the cluster. For example, if the traffic rate is high, then the values in the Total I/O Rate column in the Interfaces by Hosts table for private interfaces are also high. If the values in this column are high, then you should determine the cause of the high network usage. You can click a number to access the Network Interface Total I/O Rate page for historic statistics and metric values.

Using the Interfaces by Hosts table, you can drill down to the following pages:

- Host Home
- Hardware Details
- Network Interface Total I/O Rate
- Network Interface Total Error Rate

Using the Interfaces in Use by Cluster Databases table, you can view the Total Transfer Rate. This value shows the network traffic generated by individual instances for the interfaces they are using as interconnects. The values indicate how frequently the instances are communicating with other instances.

See Also:

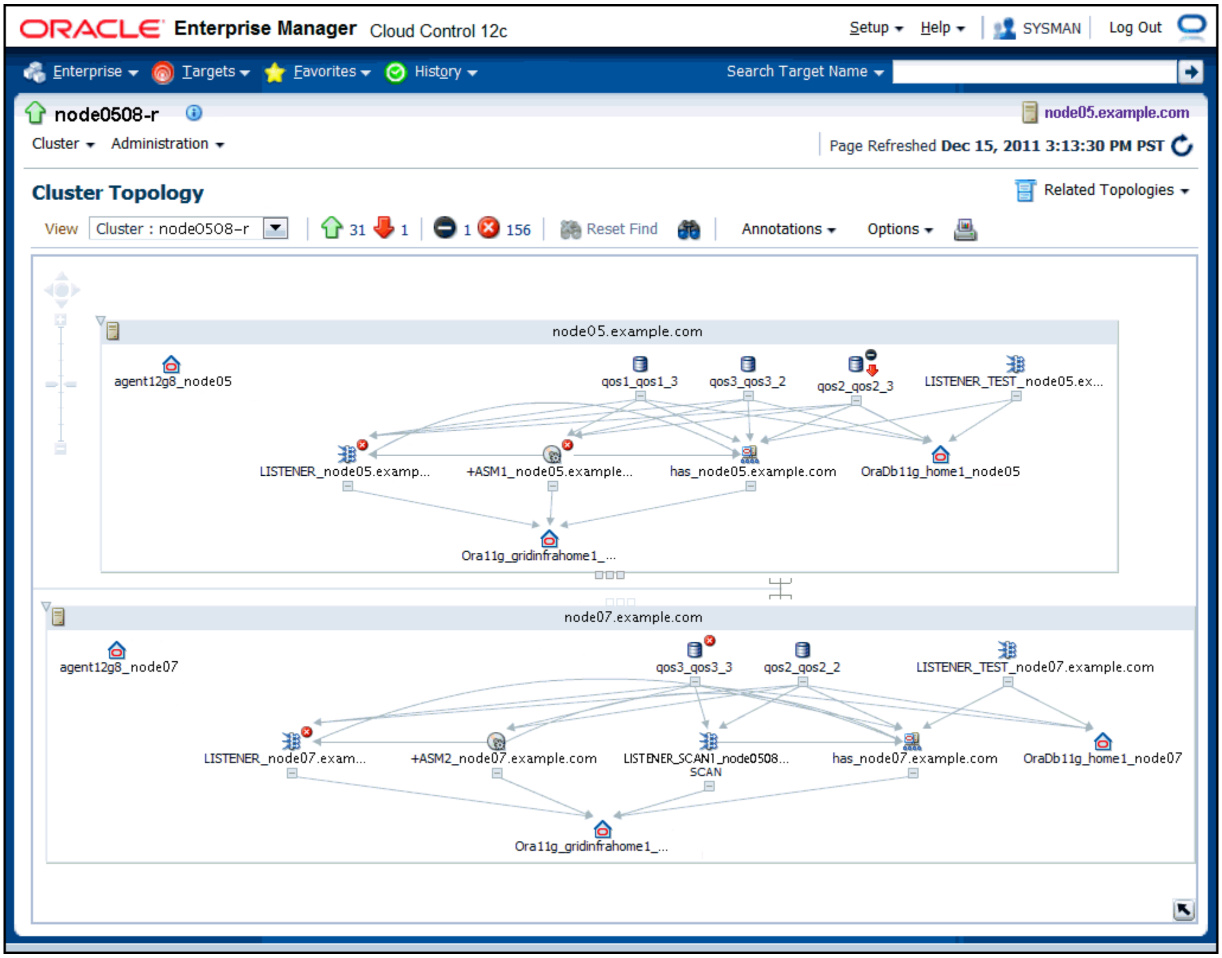
- "Monitoring Oracle Clusterware" on page 8-22
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC" on page 1-3
- "About Network Hardware Requirements" on page 2-4

Viewing the Cluster Topology Page

The Oracle Enterprise Manager Topology Viewer enables you to visually see the relationships between target types in your cluster. You can zoom in or out, pan, and see selection details. The Topology Viewer uses individually distinct icons to represent system target types, and standardized visual indicators, such as frames around selections, are used across all target types.

The Topology Viewer populates icons based on your system configuration. If a listener is serving an instance, then a line connects the listener icon and the instance icon. If a cluster database is configured to use Oracle ASM, then the relationship between the cluster Oracle ASM and cluster database appears in the topology, as shown in the following figure:

Figure 8–3 Cluster Topology Page of Oracle Enterprise Manager



If the Show Configuration Details option is unchecked, then the topology shows the monitoring view of the environment, which includes general information such as alerts and overall status. If you select the Show Configuration Details option, then additional details are shown in the Selection Details page, which are valid for any topology view. For instance, the listener component would also show the computer name and port number.

You can click an icon and then the right mouse button to display a menu of available actions. Several actions go to pages related to the target type where you can perform monitoring or tuning tasks.

Refer to the Enterprise Manager Online Help for more information about the contents of this page.

See Also: "Viewing the Cluster Database Topology" on page 8-21

Troubleshooting Configuration Problems in Oracle RAC Environments

Problems can occur when attempting to complete the installation or database creation process manually instead of using the Oracle Database management tools. Other problems occur due to the database administrator or system administrator missing important operating system or cluster configuration steps before installation. Both Oracle Clusterware and Oracle Database components have subcomponents that you can troubleshoot. The Cluster Ready Services Control (CRSCTL) command `check` enables you to determine the status of several Oracle Clusterware components simultaneously.

This section contains the following topics:

- Using CRSCTL to Diagnose Cluster Issues
- Using the Cluster Verification Utility to Diagnose Problems
- Viewing Oracle RAC Database Alerts
- Viewing Oracle RAC Database Alert Log Messages

See Also:

- "Tools for Installing, Configuring, and Managing Oracle RAC"
- "About Verifying the Oracle Clusterware Installation"

Using CRSCTL to Diagnose Cluster Issues

You can use CRSCTL commands as the `root` operating system user to diagnose problems with your Oracle Clusterware installation, or to enable dynamic debugging for Oracle Clusterware. This section contains the following topics:

- About the Oracle Clusterware Alert Log
- About the Oracle Clusterware Component Log Files
- Checking the Status of the Oracle Clusterware Installation
- Running the Oracle Clusterware Diagnostics Collection Script
- Enabling Debugging of Oracle Clusterware Components
- Enabling Debugging for an Oracle Clusterware Resource
- Enabling and Disabling Oracle Clusterware Daemons

See Also:

- "Tools for Installing, Configuring, and Managing Oracle RAC"
- "Troubleshooting Configuration Problems in Oracle RAC Environments"
- "Monitoring Oracle Clusterware"

About the Oracle Clusterware Alert Log

Oracle Clusterware posts alert messages when important events occur. For example, you might see alert messages from the Cluster Ready Services (CRS) daemon process when it starts, if it aborts, if the failover process fails, or if automatic restart of an Oracle Clusterware resource failed.

Enterprise Manager monitors the Clusterware log file and posts an alert on the Cluster Home page if an error is detected. For example, if a voting disk is not available, then a `CRS-1604` error is raised, and a critical alert is posted on the Cluster Home page. You can customize the error detection and alert settings on the Metric and Policy Settings page.

The Oracle Clusterware alert log should be the first place to look for serious errors. It often contains references to other diagnostic logs that provide detailed information on a specific component. The location of the Oracle Clusterware log file is `ORACLE_BASE/diag/crs/`*`hostname`*`/crs/trace/alert.log`, where `ORACLE_BASE` is the Oracle base path you specified when you installed Oracle Grid Infrastructure and *`hostname`* is the name of the host.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for more information about Oracle Clusterware diagnostic and alert log data
- "Monitoring Oracle Clusterware"

About the Oracle Clusterware Component Log Files

Starting with Oracle Clusterware 12*c* release 1 (12.1.0.2), diagnostic data files written by Oracle Clusterware programs are known as trace files and have a `.trc` file extension, and appear together in the `trace` subdirectory of the ADR home. Besides trace files, the `trace` subdirectory in the Oracle Clusterware ADR home contains the simple text Oracle Clusterware alert log. It always has the name `alert.log`. The alert log is also written as an XML file in the alert subdirectory of the ADR home, but the text alert log is most easily read.

See Also:

- *Oracle Clusterware Administration and Deployment Guide*
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "Monitoring Oracle Clusterware"

Checking the Status of the Oracle Clusterware Installation

Use the CRSCTL `check` command to display the status of an Oracle Clusterware component or daemon.

To determine the condition of your clusterware installation:

1. Log in to the operating system as the `root` user in a command window.

2. Use CRSCTL to check the status of Oracle Clusterware using the following command:

   ```
   # crsctl check crs
   ```

3. Check the status of an individual Oracle Clusterware daemon using the following syntax, where *daemon* is `crsd`, `cssd`, or `evmd`:

   ```
   # crsctl check daemon
   ```

4. To list the status of all Oracle Clusterware resources running on any node in the cluster, use the following command:

   ```
   # crsctl status resource -t
   ```

   This command lists the status of all registered Oracle Clusterware resources, which includes the VIPs, listeners, databases, services, and Oracle ASM instances and disk groups.

   See Also:

   - *Oracle Clusterware Administration and Deployment Guide*
   - "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
   - "Monitoring Oracle Clusterware"

Running the Oracle Clusterware Diagnostics Collection Script

The Oracle Clusterware Diagnostics Collection script collects diagnostic information for your Oracle Clusterware installation. The diagnostics provide additional information so that Oracle Support Services can resolve problems. It displays the status of the Cluster Synchronization Services (CSS), Event Manager (EVM), and the Oracle Clusterware (CRS) daemons.

To run the Oracle Clusterware Diagnostics Collection script:

1. In a command window, log in to the operating system as the `root` user.
2. Run the `diagcollection.pl` script from the operating system prompt as follows, where *Grid_home* is the home directory of your Oracle Grid Infrastructure for a cluster installation:

   ```
   # Grid_home/bin/diagcollection.pl --collect
   ```

   See Also:

   - *Oracle Clusterware Administration and Deployment Guide*
   - "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
   - "Monitoring Oracle Clusterware"

Enabling Debugging of Oracle Clusterware Components

You can enable debugging for the Oracle Cluster daemons, Event Manager (EVM), and their modules by running CRSCTL commands.

To enable debugging of Oracle Clusterware components:

1. In a command window, log in to the operating system as the `root` user.

2. Use the following command to obtain the module names for a component, where *component_name* is crs, evm, css or the name of the component for which you want to enable debugging:

   ```
   # crsctl lsmodules component_name
   ```

   For example, viewing the modules of the css component might return the following results:

   ```
   # crsctl lsmodules css
   The following are the CSS modules ::
   CSSD
   COMMCRS
   COMMNS
   ```

3. Use CRSCTL as follows, where *component_name* is the name of the Oracle Clusterware component for which you want to enable debugging, *module* is the name of module, and *debugging_level* is a number from 1 to 5:

   ```
   # crsctl debug log component module:debugging_level
   ```

   For example, to enable the lowest level of tracing for the CSSD module of the css component, you would use the following command:

   ```
   # crsctl debug log css CSSD:1
   ```

See Also:

- *Oracle Clusterware Administration and Deployment Guide*
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "Monitoring Oracle Clusterware"

Enabling Debugging for an Oracle Clusterware Resource

You can use CRSCTL commands to enable debugging for resource managed by Oracle Clusterware.

To enable debugging of an Oracle Clusterware resource:

1. In a command window, log in to the operating system as the root user.
2. Obtain a list of the resources available for debugging by running the following command:

   ```
   # crsctl check crs
   ```

3. Run the following command to enable debugging, where *resource_name* is the name of an Oracle Clusterware resource, such as ora.racnode1.vip, and *debugging_level* is a number from 1 to 5:

   ```
   # crsctl debug log res resource_name:debugging_level
   ```

See Also:

- *Oracle Clusterware Administration and Deployment Guide*
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "Monitoring Oracle Clusterware"

Enabling and Disabling Oracle Clusterware Daemons

When the Oracle Clusterware daemons are enabled, they start automatically when the node is started. To prevent the daemons from starting automatically, you can disable them using `crsctl` commands.

To enable automatic startup for all Oracle Clusterware daemons:

1. In a command window, log in to the operating system as the `root` user.
2. Run the following CRSCTL command:

   ```
   # crsctl enable crs
   ```

To disable automatic startup for all Oracle Clusterware daemons:

1. In a command window, log in to the operating system as the `root` user.
2. Run the following CRSCTL command:

   ```
   # crsctl disable crs
   ```

See Also:

- *Oracle Clusterware Administration and Deployment Guide*
- "About Oracle Grid Infrastructure for a Cluster and Oracle RAC"
- "Monitoring Oracle Clusterware"
- *Oracle Real Application Clusters Administration and Deployment Guide*

Using the Cluster Verification Utility to Diagnose Problems

The Cluster Verification Utility (CVU) can assist you in diagnosing a wide variety of configuration problems.

This section contains the following topics:

- Verifying the Existence of Node Applications
- Verifying the Integrity of Oracle Clusterware Components
- Verifying the Integrity of the Oracle Cluster Registry
- Verifying the Integrity of Your Entire Cluster
- Checking the Settings for the Interconnect
- Enabling Tracing

See Also:

- *Oracle Clusterware Administration and Deployment Guide*
- "Monitoring Oracle Clusterware"
- "Viewing the Cluster Topology Page"

Verifying the Existence of Node Applications

You use the CVU `comp nodeapp` command to verify the existence of node applications, namely the virtual IP (VIP), Oracle Notification Services (ONS), and Global Service Daemon (GSD), on all the nodes.

To verify the existence of node applications:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. Use the `comp nodeapp` command of CVU, using the following syntax:

   ```
   cluvfy comp nodeapp [ -n node_list] [-verbose]
   ```

   where *`node_list`* represents the nodes to check.
3. If the `cluvfy` command returns the value of `UNKNOWN` for a particular node, then CVU cannot determine whether a check passed or failed. Determine if the failure was caused by one of the following reasons:
   - The node is down.
   - Executable files that CVU requires are missing in the *`CRS_home`*`/bin` directory or the *`Oracle_home`*`/bin` directory.
   - The user account that ran CVU does not have permissions to run common operating system executable files on the node.
   - The node is missing an operating system patch or required package.
   - The kernel parameters on that node were not configured correctly and CVU cannot obtain the operating system resources required to perform its checks.

   See Also:

   - "Troubleshooting Configuration Problems in Oracle RAC Environments"
   - *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility

Verifying the Integrity of Oracle Clusterware Components

You use the CVU `comp crs` command to verify the existence of all the Oracle Clusterware components.

To verify the integrity of Oracle Clusterware components:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. Use the `comp crs` command of CVU, using the following syntax:

   ```
   cluvfy comp crs [ -n node_list] [-verbose]
   ```

   where *`node_list`* represents the nodes to check.

   See Also:

   - "Troubleshooting Configuration Problems in Oracle RAC Environments"
   - *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility

Verifying the Integrity of the Oracle Cluster Registry

You use the CVU `comp ocr` command to verify the integrity of the Oracle Clusterware registry.

To verify the integrity of the Oracle Clusterware registry:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. Use the `comp ocr` command of CVU, using the following syntax:

   ```
   cluvfy comp ocr [ -n node_list] [-verbose]
   ```

   where *`node_list`* represents the nodes to check.

   See Also:

   - "Troubleshooting Configuration Problems in Oracle RAC Environments"
   - *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility

Verifying the Integrity of Your Entire Cluster

You use the CVU `comp clu` command to check that all nodes in the cluster have the same view of the cluster configuration.

To verify the integrity of your cluster:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. Use the `comp clu` command of CVU, using the following syntax:

   ```
   cluvfy comp clu [-verbose]
   ```

   See Also:

   - "Troubleshooting Configuration Problems in Oracle RAC Environments"
   - *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility
   - "Verifying Your Oracle RAC Database Installation"

Checking the Settings for the Interconnect

Cache Fusion enhances the performance of Oracle RAC by utilizing a high-speed interconnect to send data blocks to another instance's buffer cache. The high-speed interconnect should be a private network with the highest bandwidth to maximize performance.

For network connectivity verification, CVU discovers all the available network interfaces if you do not specify an interface on the CVU command line.

To check the settings for the interconnect:

1. In a command window, log in to the operating system as the user who owns the Oracle Clusterware software installation.
2. To verify the accessibility of the cluster nodes, specified by *`node_list`*, from the local node or from any other cluster node, specified by *`srcnode`*, use the component verification command `nodereach` as follows:

   ```
   cluvfy comp nodereach -n node_list [ -srcnode node ] [-verbose]
   ```

3. To verify the connectivity among the nodes, specified by *node_list*, through the available network interfaces from the local node or from any other cluster node, use the comp nodecon command as shown in the following example:

   ```
   cluvfy comp nodecon -n node_list -verbose
   ```

   When you issue the nodecon command as shown in the previous example, it instructs CVU to perform the following tasks:

   - Discover all the network interfaces that are available on the specified cluster nodes.
   - Review the corresponding IP addresses and subnets for the interfaces.
   - Obtain the list of interfaces that are suitable for use as VIPs and the list of interfaces to private interconnects.
   - Verify the connectivity among all the nodes through those interfaces.

   When you run the nodecon command in verbose mode, it identifies the mappings between the interfaces, IP addresses, and subnets.

4. To verify the connectivity among the nodes through specific network interfaces, use the comp nodecon command with the -i option and specify the interfaces to be checked with the *interface_list* argument:

   ```
   cluvfy comp nodecon -n node_list -i interface_list [-verbose]
   ```

   For example, you can verify the connectivity among the nodes racnode1, racnode2, and racnode3, through the specific network interface eth0 by running the following command:

   ```
   cluvfy comp nodecon -n racnode1,racnode2,racnode3 -i eth0 -verbose
   ```

See Also:

- "Troubleshooting Configuration Problems in Oracle RAC Environments"
- "Configuring the Network"
- "Configuring Secure Shell on Linux Systems"
- *Oracle Clusterware Administration and Deployment Guide* for more information about the Cluster Verification Utility

Enabling Tracing

CVU does not generate trace files unless you enable tracing. The CVU trace files are created in the *CRS_home*/cv/log directory. Oracle RAC automatically rotates the log files, and the most recently created log file has the name cvutrace.log.0. You should remove unwanted log files or archive them to reclaim disk space, if needed.

To enable tracing using CVU:

1. In a command window, log in to the operating system as the root user.
2. Set the environment variable SRVM_TRACE to true.

   ```
   # set SRVM_TRACE=true; export SRVM_TRACE
   ```

3. Run the command to trace.

See Also:

- "Troubleshooting Configuration Problems in Oracle RAC Environments"
- *Oracle Clusterware Administration and Deployment Guide* for more information about enabling tracing

Viewing Oracle RAC Database Alerts

Alert messages are displayed in Enterprise Manager. The Alerts table is similar to that shown for single-instance databases, but in a cluster database, it includes columns for the target name and target type. For example, if a user connected to the `orcl1` instance exceeded their allotted login time, then you would see an alert message with the following values:

- Target name: `orcl_orcl1`
- Target type: Database instance
- Category: Response
- Name: User logon time
- Message: User logon time is 10250 milliseconds
- Alert triggered: *Date and time when the alert condition occurred*

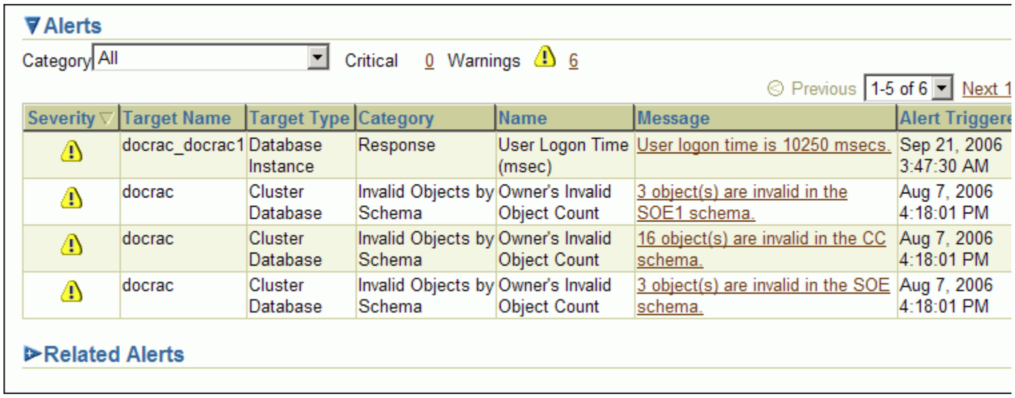
To view the alert messages for an Oracle RAC database:

1. On the Cluster Database Home page, scroll down to the section titled Alerts.

   The section Related Alerts displays nondatabase alert messages, for example, alert messages for Oracle Net.

2. View the alerts for your database and database instances.

   The following screenshot shows an example of the Alerts display for a clustered database named `docrac`.



Category All

Critical 0 Warnings ⚠ 6

Severity	Target Name	Target Type	Category	Name	Message	Alert Triggere
⚠	docrac_docrac1	Database Instance	Response	User Logon Time (msec)	User logon time is 10250 msecs.	Sep 21, 2006 3:47:30 AM
⚠	docrac	Cluster Database	Invalid Objects by Schema	Owner's Invalid Object Count	3 object(s) are invalid in the SOE1 schema.	Aug 7, 2006 4:18:01 PM
⚠	docrac	Cluster Database	Invalid Objects by Schema	Owner's Invalid Object Count	16 object(s) are invalid in the CC schema.	Aug 7, 2006 4:18:01 PM
⚠	docrac	Cluster Database	Invalid Objects by Schema	Owner's Invalid Object Count	3 object(s) are invalid in the SOE schema.	Aug 7, 2006 4:18:01 PM

See Also:

- "Monitoring Oracle RAC Database and Cluster Performance"
- "Troubleshooting Configuration Problems in Oracle RAC Environments"
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about monitoring performance alerts

Viewing Oracle RAC Database Alert Log Messages

An alert log is created for each instance in a cluster database.

To view the alert log for an Oracle RAC database instance:

1. On the Cluster Database Home page, scroll down to the Instances section.
2. Click the name of the instance for which you want to view the alert log.

   The Cluster Database Instance Home page appears.
3. In the Diagnostic Summary section, click the link next to the heading **Alert Log** to display the alert log entries containing ORA- errors.

   The Alert Log Errors page appears.
4. (Optional) Click **Alert Log Contents** in the Related Links section to view all the entries in the alert log.

   On the View Alert Log Contents page, click **Go** to view the most recent entries, or you can enter your own search criteria.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about where to find files for analyzing errors in Oracle RAC
- "Monitoring Oracle Clusterware"

9 Adding and Deleting Nodes and Instances

You can extend an existing Oracle Real Application Clusters (Oracle RAC) home to other nodes and instances in the cluster, and delete Oracle RAC instances and remove nodes from the cluster. You can add nodes when configuring a new Oracle RAC cluster, or you can remove nodes to resize an existing cluster.

> **Note:** When adding or deleting nodes, it is very important that you perform each step in the order shown.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

Preparing the New Node

Before a node can be added to the cluster, you must perform the same preinstallation steps on the new node as you did for all the existing nodes in the cluster. This includes the following tasks:

- Checking hardware compatibility
- Configuring the operating system
- Configuring SSH connectivity between the new node and the other cluster members
- Configuring access to shared storage
- Creating groups, users, and directories

See Also:

- "Preparing Your Cluster" on page 2-1
- "Configuring the Network" on page 2-16
- "About Verifying the Oracle Clusterware Installation" on page 3-19

Verifying the New Node Meets the Prerequisites for Installation

When adding a node to an existing cluster, the new node must match the configuration of the other nodes in the cluster. The new nodes must run the same operating system and version of the operating system as the other nodes. The new computer must also use the same chip architecture (32-bit or 64-bit) as the other nodes. However, you can have computers of different speeds and sizes in the same cluster.

After you have configured the new nodes, use `cluvfy` to verify that all the requirements for installation have been met. To verify the new node meets the hardware requirement, run the following command on an existing node (for example, either `racnode1` or `racnode2`) from the *`Grid_home`*`/bin` directory:

```
cluvfy stage -pre crsinst -n racnode3 -verbose
```

See Also:

- "Preparing Your Cluster" on page 2-1
- "Configuring the Network" on page 2-16
- "About Verifying the Oracle Clusterware Installation" on page 3-19

Extending the Oracle Grid Infrastructure Home to the New Node

Now that the new node has been configured to support Oracle Clusterware, you use Oracle Universal Installer (OUI) to add a Grid home to the node being added to your cluster.

This section assumes that you are adding a node named `racnode3` and that you have successfully installed Oracle Clusterware on `racnode1` in a nonshared home, where *`Grid_home`* represents the successfully installed Oracle Clusterware home.

To extend the Oracle Grid Infrastructure for a cluster home to include the new node:

1. Verify the new node has been properly prepared for an Oracle Clusterware installation by running the following CLUVFY command on the `racnode1` node:

   ```
   cluvfy stage -pre nodeadd -n racnode3 -verbose
   ```

2. As the `oracle` user (owner of the software installation for Oracle Grid Infrastructure for a cluster) on `racnode1`, go to *`Grid_home`*`/addnode` and run the `addNode.sh` script in silent mode:

   If you are using Grid Naming Service (GNS):

   ```
   ./addNode.sh -silent "CLUSTER_NEW_NODES={racnode3}"
   ```

   If you are not using Grid Naming Service (GNS):

   ```
   ./addNode.sh -silent "CLUSTER_NEW_NODES={racnode3}" "CLUSTER_NEW_VIRTUAL_
   ```

```
HOSTNAMES={racnode3-vip}"
```

When running this command, the curly braces ({ }) are not optional and must be included or the command returns an error.

You can alternatively use a response file instead of placing all the arguments in the command line. See *Oracle Clusterware Administration and Deployment Guide* for more information on using response files.

3. When the script finishes, run the `root.sh` script as the `root` user on the new node, `racnode3`, from the Oracle home directory on that node.
4. If you are not using Oracle Grid Naming Service (GNS), then you must add the name and address for `racnode3` to DNS.

You should now have Oracle Clusterware running on the new node. To verify the installation of Oracle Clusterware on the new node, you can run the following command on the newly configured node, `racnode3`:

```
$ cd /u01/app/12.1.0/grid/bin
$ ./cluvfy stage -post nodeadd -n racnode3 -verbose
```

> **Note:** Avoid changing host names after you complete the Oracle Clusterware installation, including adding or deleting domain qualifications. Nodes with changed host names must be deleted from the cluster and added back with the new name.

See Also:

- "Completing the Oracle Clusterware Configuration"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database
- *Oracle Clusterware Administration and Deployment Guide* for more information on using response files

Extending the Oracle RAC Home Directory

Now that you have extended the Grid home to the new node, you must extend the Oracle home on `racnode1` to `racnode3`. The following steps assume that you have completed the tasks described in the previous sections, "Preparing the New Node" and "Extending the Oracle Grid Infrastructure Home to the New Node", and that `racnode3` is a member node of the cluster to which `racnode1` belongs.

The procedure for adding an Oracle home to the new node is very similar to the procedure you just completed for extending the Grid home to the new node.

To extend the Oracle RAC installation to include the new node:

1. Ensure that you have successfully installed the Oracle RAC software on at least one node in your cluster environment. To use these procedures as shown, replace *`Oracle_home`* with the location of your installed Oracle home directory.
2. Go to the *`Oracle_home`*`/addnode` directory on `racnode1` and run the `addNode.sh` script in silent mode as shown in the following example:

```
$ cd /u01/app/oracle/product/12.1.0/dbhome_1/addnode
$ ./addNode.sh -silent "CLUSTER_NEW_NODES={racnode3}"
```

3. When the script finishes, run the `root.sh` script as the `root` user on the new node, `racnode3`, from the Oracle home directory on that node.

   For policy-managed databases with Oracle Managed Files (OMF) enabled, no further actions are needed.

   For a policy-managed database, when you add a new node to the cluster, it is placed in the Free pool by default. If you increase the cardinality of the database server pool, then an Oracle RAC instance is added to the new node, `racnode3`, and it is moved to the database server pool. No further action is necessary.

4. Add shared storage for the undo tablespace and redo log files.

   If OMF is not enabled for your database, then you must manually add an undo tablespace and redo logs.

5. If you have an administrator-managed database, then add a new instance on the new node as described in "Creating an Instance on the New Node" on page 9-4.

   If you followed the installation instructions in this guide, then your cluster database is an administrator-managed database and stores the database files on Oracle Automatic Storage Management (Oracle ASM) with OMF enabled.

After completing these steps, you should have an installed Oracle home on the new node.

See Also:

- "Verifying Your Oracle RAC Database Installation"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

Adding the New Node to the Cluster using Enterprise Manager

If you followed the steps described in "Extending the Oracle RAC Home Directory", then the cluster node is added to the cluster by the `addNode.sh` script. After the software is started on the new node, it is detected by Oracle Enterprise Manager. If an Oracle Enterprise Manager agent was not installed on the new node, then an alert is issued for that host, with the message "Incomplete configuration."

See Also:

- "Completing the Oracle Clusterware Configuration"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

Creating an Instance on the New Node

You can add an instance to the cluster using either the Instance Management option of Database Configuration Assistant (DBCA) or using Enterprise Manager. Before using either of these options you must first configure the new node to be a part of the cluster and install the software on the new node as described in the previous sections.

There are two methods of adding an instance to the new node:

- Adding a New Instance for a Policy-Managed Database

- Adding a New Instance for an Administrator-Managed Database

Note: The steps described in this section require a license for the Enterprise Manager Provisioning Management pack. Refer to the *Oracle Database Licensing Information* for information about the availability of these features on your system.

See Also:

- "About Oracle Real Application Clusters"
- "Extending the Oracle Grid Infrastructure Home to the New Node"
- "Extending the Oracle RAC Home Directory"

Adding a New Instance for a Policy-Managed Database

To add an instance to a policy-managed database, you simply increase the cardinality of the server pool for the database. The database instance and Oracle ASM instance on the new node are created and configured automatically when a node is added to the server pool.

To add an instance to a policy-managed database using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Cluster Database Operations**, and then select **Add Instance**.

   The Add Instance: Cluster Credentials page appears.

3. Enter the credentials of the Oracle Clusterware software owner and the SYSASM user, then click **Next**.

   After the credentials have been validated, the Edit Server Pool page appears.

4. Modify the server pool settings then click **Submit**.

5. From the **Cluster Database** menu, select **Home** to return to the Cluster Database Home page.

   Review the number of instances available for your cluster database.

See Also:

- "Creating an Instance on the New Node" on page 9-4
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

Adding a New Instance for an Administrator-Managed Database

When adding an instance to an administrator-managed database, you must specify the name of the database instance and which node it should run on.

To add an instance to an administrator-managed database using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Cluster Database Operations**, and then select **Add Instance**.

   The Add Instance: Cluster Credentials page appears.

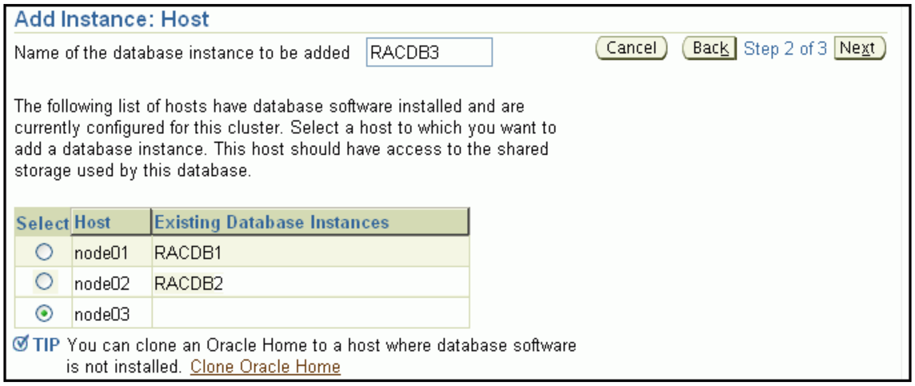
3. Enter the credentials of the Oracle Clusterware software owner and for the SYSASM user, then click **Next**.

   After the credentials have been validated, the Add Instance: Host page appears.

4. In the **Name of the database instance to be added** field, either use the default instance name, or enter a different name for the database instance, such as `racnode3`.

5. Select the node on which you want to create the new instance, then **Next**.

   > **Note:** This procedure assumes that the Oracle Database software is configured on the selected node and that there is no instance for the cluster database currently running on the selected node.



Add Instance: Host

Name of the database instance to be added: RACDB3

The following list of hosts have database software installed and are currently configured for this cluster. Select a host to which you want to add a database instance. This host should have access to the shared storage used by this database.

Select	Host	Existing Database Instances
○	node01	RACDB1
○	node02	RACDB2
●	node03	

TIP You can clone an Oracle Home to a host where database software is not installed. Clone Oracle Home

   After the selected host has been validated, the Add Instance: Review page appears.

6. Review the information, then click **Submit Job** to proceed.

   A confirmation page appears.

7. Click **View Job** to check on the status of the submitted job.

   The Job Run detail page appears.

8. Click your browser's Refresh button until the job shows a status of Succeeded or Failed.

   If the job shows a status of Failed, then you can click the name of the step that failed to view the reason for the failure.

9. From the **Cluster Database** menu, select **Home** to return to the Cluster Database Home page.

The number of instances available in the cluster database is increased by one.

See Also:

- "Creating an Instance on the New Node"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

Deleting an Instance From the Cluster Database

Sometimes, it might be necessary to remove a database instance from your cluster database. For example, to retire or repurpose a server, you first remove the database instance running on that server.

You can delete an instance from the cluster using either the Instance Management option of Database Configuration Assistant (DBCA) or using Enterprise Manager. If you choose to use DBCA to delete an instance, then start the DBCA utility from a node that will remain part of the cluster.

When you delete an instance from a clustered database, there are two methods:

- Deleting an Instance From a Policy-Managed Database
- Deleting an Instance From an Administrator-Managed Database

---

Note: The steps described in this section require a license for the Enterprise Manager Provisioning Management pack. Refer to the *Oracle Database Licensing Information* for information about the availability of these features on your system.

See Also:

- "About Oracle Real Application Clusters"
- "Adding the New Node to the Cluster using Enterprise Manager"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about removing Oracle RAC from a cluster node

Deleting an Instance From a Policy-Managed Database

To delete an instance from a policy-managed databases, you simply decrease the cardinality of the server pool for the database. The database instance and Oracle ASM instance on the deallocated node are removed from the cluster and the node is either reassigned to another server pool or placed in the Free pool.

To delete a policy-managed database instance using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

2. From the **Availability** menu, select **Cluster Database Operations**, and then select **Delete Instance**.

   The Edit Server Pool page appears.

3. Enter the credentials for the Oracle Database software owner and the SYSASM user, then click **Next**.

   The Delete Instance: Database Instance page appears

4. Modify the server pool settings then click **Submit**.

5. From the **Cluster Database** menu, select **Home** to return to the Cluster Database Home page.

   Review the number of instances available for your cluster database.

   See Also:

   - "Deleting an Instance From the Cluster Database"
   - *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

Deleting an Instance From an Administrator-Managed Database

To remove an instance from an administrator-managed database, you must specify the name of the database instance to delete and which node the instance is running on.

To delete an administrator-managed database instance using Enterprise Manager:

1. In Oracle Enterprise Manager, go to the Cluster Database Home page.

   See *Oracle Database 2 Day DBA* for details on logging in to Oracle Enterprise Manager.

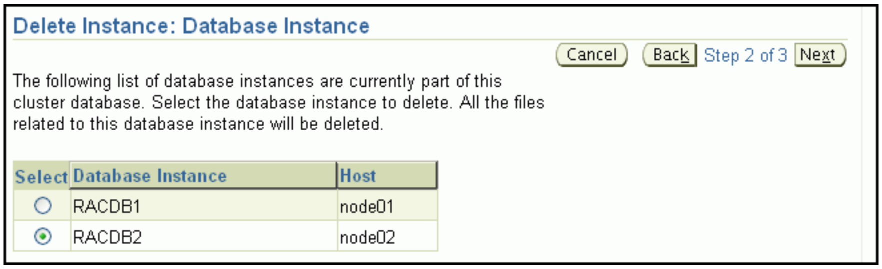
2. From the **Availability** menu, select **Cluster Database Operations**, and then select **Delete Instance**.

   The Delete Instance: Cluster Credentials page appears.

3. Enter the credentials for the Oracle Database software owner and the SYSASM user, then click **Next**.

   The Delete Instance: Database Instance page appears

4. Select the instance you want to delete, then click **Next**.



Delete Instance: Database Instance

The following list of database instances are currently part of this cluster database. Select the database instance to delete. All the files related to this database instance will be deleted.

Select	Database Instance	Host
○	RACDB1	node01
◉	RACDB2	node02

   After the host information has been validated, the Delete Instance: Review page appears.

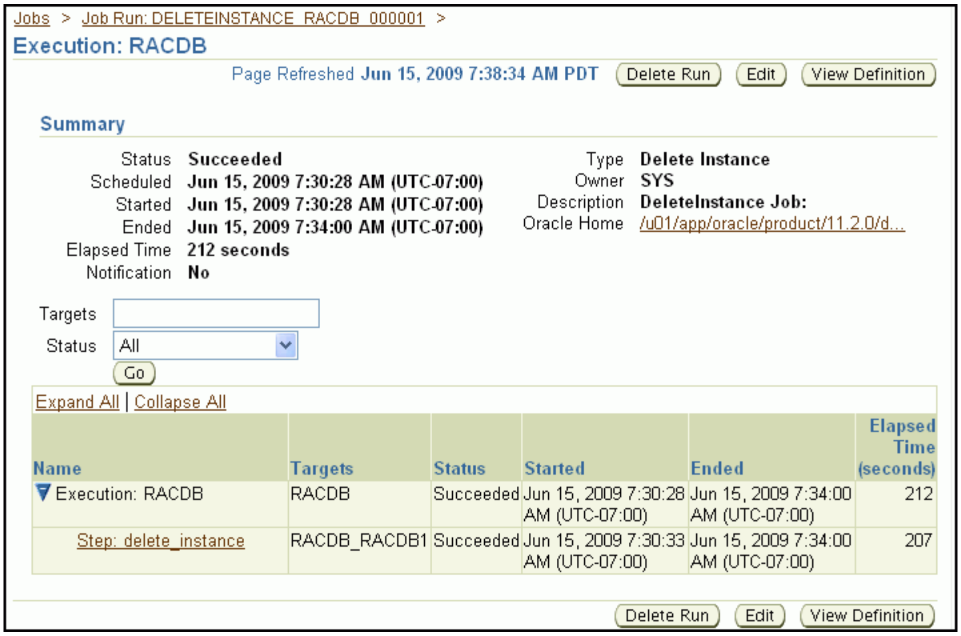
5. Review the information, and if correct, click **Submit Job** to continue. Otherwise, click **Back** and correct the information.

   A Confirmation page appears.

6. Click **View Job** to view the status of the node deletion job.

A Job Run detail page appears.

7. Click your browser's Refresh button until the job shows a status of Succeeded or Failed.


Jobs > Job Run: DELETEINSTANCE_RACDB_000001 >
Delete Run
Edit
View Definition
Go
Expand All | Collapse All
Delete Run
Edit
View Definition

Execution: RACDB

Page Refreshed **Jun 15, 2009 7:38:34 AM PDT**

Summary

Status	**Succeeded**	Type	**Delete Instance**
Scheduled	**Jun 15, 2009 7:30:28 AM (UTC-07:00)**	Owner	**SYS**
Started	**Jun 15, 2009 7:30:28 AM (UTC-07:00)**	Description	**DeleteInstance Job:**
Ended	**Jun 15, 2009 7:34:00 AM (UTC-07:00)**	Oracle Home	/u01/app/oracle/product/11.2.0/d...
Elapsed Time	**212 seconds**		
Notification	**No**		

Targets

Status All

Name	Targets	Status	Started	Ended	Elapsed Time (seconds)
▼ Execution: RACDB	RACDB	Succeeded	Jun 15, 2009 7:30:28 AM (UTC-07:00)	Jun 15, 2009 7:34:00 AM (UTC-07:00)	212
Step: delete_instance	RACDB_RACDB1	Succeeded	Jun 15, 2009 7:30:33 AM (UTC-07:00)	Jun 15, 2009 7:34:00 AM (UTC-07:00)	207

If the job shows a status of Failed, then you can click the name of the step that failed to view the reason for the failure.

8. From the **Cluster Database** menu, select **Home** to return to the Cluster Database Home page.

   The number of available instances in the cluster database is reduced by one.

See Also:

- "Deleting an Instance From the Cluster Database"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

Removing a Node From the Cluster

Removing a node from the cluster can be as easy as simply shutting down the server. If the node was not pinned and does not host any Oracle databases using Oracle Database 11*g* release 1 or earlier, then the node is automatically removed from the cluster when it is shut down. If the node was pinned or if it hosts a database instance from earlier releases, then explicit deletion is needed.

To repurpose this server, you can restart the node with a different profile in place, or you can use other software to put a new software image on the server.

See Also:

- "Completing the Oracle Clusterware Configuration"
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

10 Managing Oracle Software and Applying Patches

To manage Oracle software and apply patches in Oracle Real Application Clusters (Oracle RAC) environments, you use Oracle Universal Installer or the OPatch utility.

> **See Also:**
>
> - *Oracle Database 2 Day DBA* for more information about software management and patch releases
> - *Oracle OPatch User's Guide for Windows and UNIX* for details on how to get the latest version of OPatch

About Patches and Patch Sets

Oracle issues product fixes for its software called **patches**. They are associated with particular releases and versions of Oracle products. The patching cycle involves downloading patches, applying patches, and verifying the applied patch to ensure that the bug fixes present in the patch reflect appropriately.

Patching involves migrating from one version of the software product to another, within a particular release, unlike upgrading which involves moving from one release of a product to another newer release of the software product. When you apply the patch to your Oracle software installation, it updates the executable files, libraries, and object files in the software home directory. The patch application can also update configuration files and Oracle-supplied SQL schemas.

Oracle periodically releases the following types of patches to fix the bugs encountered in the Oracle software:

Patch Type	Description
Interim Patches	Released to fix a bug, or a collection of bugs. Previously called patch set exceptions (PSE), one-off patches, or hot fixes.
Interim Patches (for Security bug fixes)	Released to provide customer specific security fixes. Previously referred to as a test patch, fix verification binary, or e-fix.
Diagnostic Patches	Mainly help diagnose and verify a fix, or a collection of bug fixes.
Bundle Patch Updates	Cumulative collection of fixes for a specific product or component. Previously referred to as a maintenance pack, service pack, cumulative patch, update release, or MLR.
Patch Set Updates (PSU)	Cumulative patch bundles that contain well-tested and proven bug fixes for critical issues. PSUs have limited new content, and do not include any changes that require re-certification.
Security Patch Updates	A cumulative collection of security bug fixes. Previously known as Critical Patch Updates, or CPUs.

Interim patches are bug fixes available to customers in response to specific bugs. They require a particular base release or patchset to be installed before you can apply them. These patches are not versioned and are generally available in a future patchset and also the next product release. Interim patches are applied by using Enterprise Manager Cloud Control or OPatch, which is included with your Oracle Database installation.

Patch sets updates (PSUs) and patch bundles are mechanisms for delivering fully tested and integrated product fixes. All the fixes in a patch set have been tested and are certified to work with each other. Because a patch set includes only low impact patches, it does not require you to certify applications or tools against the updated Oracle Database software. When you apply a patch set, many different files and utilities are modified. This results in a release number change for your Oracle software, for example, from Oracle Database 11.2.0.3.0 to Oracle Database 11.2.0.3.2. You use OPatch to apply PSUs.

Using Oracle Enterprise Manager Cloud Control for Patching Operations

Using Cloud Control with its Provisioning & Patching functionality, you can automate the patching of your Oracle Grid Infrastructure and Oracle RAC software. Before you can use Cloud Control to patch your Oracle software, you must perform the following system configuration tasks:

- Install the Enterprise Manager Agent on all cluster nodes
- Use PDP setup to ensure the host user has the necessary privileges to complete patching tasks
- Configure named and preferred credentials in Enterprise Manager
- Configure a software library for storing patch files

Details on how to perform these tasks, and how to patch your Oracle Grid Infrastructure and Oracle RAC software using Cloud Control are available from the following PDF file: `http://www.oracle.com/technetwork/oem/pdf/512066.pdf`

The rest of this chapter describes how to install patches without using Cloud Control.

Obtaining the Patch

You obtain patches and patch sets from My Oracle Support, which is the Oracle Support Services website, at

`https://support.oracle.com`

To locate patches on the My Oracle Support website:

1. Log in to your account on My Oracle Support.
2. Select the **Patches & Updates** tab.
3. If you know the patch number, then you can enter it into the Patch Name or Number field, then click **Search**.

   If you want to search for all available patches for your system, then select **Product or Family (Advanced Search)**, which is located above the Patch Name or Number field. Supply the following information:

   - Choose the products you want to patch (for example, Oracle Clusterware, Oracle Database, or an individual product such as Universal Installer)
   - Specify the software release for the products you selected, for example, Oracle 12.1.0.1.0.
   - Specify the platform on which the software is installed.

   Click **Search** to look for available patches. The Patch Search Results page is displayed.
4. On the Patch Search Results page, select the number of the patch you want to download. A details page for that patch appears on your screen.
5. Click the **ReadMe** button to view the ReadMe file for the patch, or click **Download** to download the patch to your local computer. If you download the patch to a computer that is not a node in the cluster, then you must transfer the file using a binary protocol to a cluster node.

   You can also choose to add the patch to your current patch plan or create a new patch plan. A *patch plan* is a collection of patches that you want to apply as a group. To learn more about using patch plans, in the Add to Plan drop-down list select **Why use a Plan?**

   See Also:

   - "Verifying My Oracle Support Credentials" on page 3-3
   - "About Downloading and Installing Oracle Database Patches" on page 3-20
   - *Oracle Database 2 Day DBA* for more information about software management and patch releases

Preparing to Use OPatch

Before you apply the patch to your Oracle RAC database, your Oracle ASM installation, or to your Oracle Clusterware installation, you must prepare the operating system environment and stage the patch locally.

The following sections describe the steps to perform in preparation for using OPatch:

- Checking the ORACLE_HOME Environment Variable
- Performing a Backup Before Patching Software

- Staging the Patch on Each Node
- Updating the PATH Environment Variable
- Configuring SSH User Equivalency

See Also:

- *Oracle Database 2 Day DBA* for more information about how to keep your Oracle Database software up-to-date
- *Oracle OPatch User's Guide for Windows and UNIX*

Checking the ORACLE_HOME Environment Variable

OPatch verifies if the Oracle home is present. You must ensure that the `ORACLE_HOME` environment variable is set to the Oracle home of the product you are trying to patch. Also, you must use a version of OPatch that has the same version number as the Oracle home being patched.

Check the respective vendor documentation for the details to set the environment variable.

To check the current setting of the `ORACLE_HOME` variable on Linux:

1. In a command window, log in to the operating system as the `oracle` user.
2. Use the `echo` command to display the current setting of the `ORACLE_HOME` environment variable.

   ```
   echo $ORACLE_HOME
   ```

See Also:

- "Preparing to Use OPatch" on page 10-3
- "Configuring the Operating System Environment" on page 3-4

Performing a Backup Before Patching Software

Oracle strongly recommends that you back up the software directory you are patching before performing any patch operation. This applies to Oracle Database or Oracle Grid Infrastructure software installation directories.

To back up the software installation:

1. Back up the software installed in the specified Oracle home using:

   a. An operating system utility, such as `zip`, `cp -r`, `tar`, or `cpio`, to back up the software in the Oracle home directory to disk.

   b. The Oracle Secure Backup utility to back up the software in the Oracle home directory to tape.

See Also:

- "Preparing to Use OPatch" on page 10-3
- "Configuring the Operating System Environment" on page 3-4

Staging the Patch on Each Node

After you download the patch from My Oracle Support, you must copy the patch to each node that runs the software being patched.

See Also:

- "Preparing to Use OPatch" on page 10-3
- *Oracle Database 2 Day DBA* for more information about software management and patch releases

Updating the PATH Environment Variable

The `opatch` binary file is located in the *`Oracle_home`*`/OPatch` directory. You can either specify this path when executing OPatch, or you can update the `PATH` environment variable to include the `OPatch` directory.

To update the PATH environment variable on Oracle Linux systems:

1. In a command window, log in to the operating system as the Oracle home owner, or a user that has write permission for the files in the Oracle home.
2. Use a shell command similar to the following to update the value of the PATH environment variable, where `/u01/app/oracle/product/12.1.0/dbhome_1` is the location of your Oracle home directory:

   ```
   $ export PATH=$PATH:/u01/app/oracle/product/12.1.0/dbhome_1/OPatch
   ```

   You could also modify the shell profile script for the current user to have this variable configured every time you log in.

See Also:

- "Preparing to Use OPatch" on page 10-3
- "Configuring the Operating System Environment" on page 3-4

Configuring SSH User Equivalency

Before you patch a system, make sure the user equivalency is working. **User equivalency** exists when the operating system user on each node is considered to be identical. For information on configuring user equivalency, see "Configuring Operating System Users and Groups on Linux Systems" on page 2-13.

To test SSH user equivalency:

1. On the system where you want to run OPatch, log in as the `oracle` user.
2. Use the following command to test user equivalency:

   ```
   [oracle@racnode1] $ ssh racnode2 date
   ```

   If the date is returned, then user equivalency between the source and destination node has been configured.
3. If you see output similar to the following, then SSH user equivalency is not enabled:

   ```
   Enter passphrase for key '/home/oracle/.ssh/id_rsa':
   ```

   Enable SSH user equivalency before continuing with the patching operation.

To enable SSH user equivalency:

1. On the system where you want to run OPatch, open a command window and log in as the `oracle` user.
2. Start the SSH agent and load the SSH keys into memory using the following commands:

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

   These commands start the `ssh-agent` on the local node, and load the RSA and DSA keys into the current session's memory so that you are not prompted to use pass phrases when issuing SSH commands.
3. At the prompt, enter the pass phrase for each key that you generated when configuring Secure Shell, for example:

   ```
   [oracle@racnode1 .ssh]$ exec /usr/bin/ssh-agent $SHELL
   [oracle@racnode1 .ssh]$ /usr/bin/ssh-add
   Enter passphrase for /home/oracle/.ssh/id_rsa
   Identity added: /home/oracle/.ssh/id_rsa (/home/oracle/.ssh/id_rsa)
   Identity added: /home/oracle/.ssh/id_dsa (/home/oracle/.ssh/id_dsa)
   ```

4. To test if you have configured SSH correctly, run the following command. If you have configured SSH correctly, then you are not prompted for a password or a pass phrase.

   ```
   [oracle@racnode1] $ ssh racnode2 date
   ```

Note: Do not close this command window until you have completed the patch installation. If you must close the command window in which you enabled SSH user equivalency before the patch installation is complete, then repeat Step 1 to Step 4 before starting the patch installation.

See Also:

- "Preparing to Use OPatch" on page 10-3
- "Configuring Operating System Users and Groups on Linux Systems" on page 2-13
- *Oracle Grid Infrastructure Installation Guide* for your specific operating system for instructions on how to configure SSH

Applying Patches

Patching in an Oracle RAC environment is slightly different compared to patching a single node. If OPatch detects a cluster, then it uses OUI to query the software inventory to find the local node name and node list.

Before you install a patch, you must stop all the applications running from the software directory that is being patched. In a cluster, you may have to shut down additional applications, depending upon which software is being patched. The following table lists the applications to stop when patching Oracle software.

Table 10–1 Patching Oracle Home Directories

Oracle Home Directory	Applications to Stop
Oracle RAC Database	Oracle RAC database, Enterprise Manager Agent, listener, and any other applications that run from the Oracle RAC home directory
Oracle Grid Infrastructure	Oracle RAC database and all applications running from the Oracle RAC home directory, any single-instance databases that use the same Oracle ASM instance as the cluster database, the Oracle ASM instance, all node applications, Oracle Clusterware, and any other applications that run from the Grid home directory

This section describes the following tasks:

- Patching the Grid Home
- All Node Patching
- Rolling Patching
- Minimum Downtime Patching

> **See Also:**
>
> - "Obtaining the Patch" on page 10-3
> - "Preparing to Use OPatch" on page 10-3
> - *Oracle Universal Installer NextGen Installation Guide*
> - *Oracle OPatch User's Guide for Windows and UNIX* for information about patch automation

Patching the Grid Home

There are two parts to patching Oracle Grid Infrastructure for a cluster:

- Unlock the Grid home
- Apply patches to the Grid home

Before you can apply a patch or make other modifications to the software files in the Grid home, you must first unlock the Grid home.

To unlock the Grid home, perform the following steps:

1. Change directory to the path `Grid_home/crs/install`, where *`Grid_home`* is the path to the Grid home, and at the `root` user unlock the Grid home using commands similar to the following:

   ```
   cd /u01/app/12.1.0/grid/crs/install
   rootcrs.bat -unlock -crshome /u01/app/12.1.0/grid
   ```

   When you run the `rootcrs.bat` script with the `-unlock` flag, it stops the Oracle Clusterware stack and unlocks the files in the Grid home so they can be modified.

> **Note:** Starting with Oracle Database 12*c* Release 1 (12.1.0.2), the `rootcrs.bat` script replaces the `rootcrs.pl` script in the Oracle Grid Infrastructure home.

2. Change user to the software owner for Oracle Grid Infrastructure for a cluster and apply the patch to the Grid home, using one of the patching methods described in this section.
3. After you have finished modifying the Grid home, lock it again as the `root` user using commands similar to the following:

   ```
   cd /u01/app/12.1.0/grid/crs/install
   rootcrs.bat -patch
   ```

   The `rootcrs.bat` script with the `-patch` flag locks the Grid home again and restarts the Oracle Clusterware stack.

All Node Patching

In all node patching, all the nodes in the cluster are initially shut down and the patch is applied on all the nodes. After all the nodes have been patched, then the Oracle Clusterware stack and all its registered resources are restarted on each node. This method is typically used for very critical patches and it leads to maximum downtime. OPatch uses this method if the patch cannot be applied in a rolling fashion, and you did not specify the `minimize_downtime` option.

See Also:

- "Obtaining the Patch" on page 10-3
- "Preparing to Use OPatch" on page 10-3
- "Applying Patches" on page 10-6
- "Applying Patch Sets" on page 10-9
- "Troubleshooting Patch Deployment" on page 10-10
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about SRVCTL commands
- *Oracle Clusterware Administration and Deployment Guide* for information about CRSCTL commands

Rolling Patching

In rolling patching, one group of nodes is shut down, the patch is applied to those nodes, and the nodes are brought back up. This is performed group by group, separately, until all the nodes in the cluster are patched. This is the most efficient means of applying an interim patch to an Oracle RAC or Oracle Grid Infrastructure for a cluster installation. By patching groups of nodes individually, there is zero downtime for the cluster database because at least one instance is always available on a different node.

While most patches can be applied in a rolling fashion, some patches cannot be applied in this fashion. The `README` file for the patch indicates whether you can apply the patch using the rolling patch method. If the patch cannot be applied using the rolling patch method, then you must use either "Minimum Downtime Patching" on page 10-9 or "All Node Patching" on page 10-8 to apply the patch.

See Also:

- "Obtaining the Patch" on page 10-3
- "Preparing to Use OPatch" on page 10-3
- "Applying Patches" on page 10-6
- "Applying Patch Sets" on page 10-9
- "Troubleshooting Patch Deployment" on page 10-10

Minimum Downtime Patching

In **minimum downtime patching**, one set of nodes is shut down and the patch is applied to those nodes. After the first set of nodes has been patched, the second set of nodes is shut down. The first set of nodes is then restarted and the patch is applied to the second set of nodes. After the patch has been applied to the second set of nodes, those nodes are restarted. This method leads to less downtime for Oracle RAC, compared to having all the nodes shut down at the same time.

When you use the minimum downtime patching method, the following actions occur:

- The local node is always patched first.
- The local node is used as a base to patch the other nodes.
- The user is prompted for the set of nodes to patch first from the remaining nodes.
- For each node in this first set, the user is asked to stop the instance and then the patch is propagated to that node before continuing to the next node. When the initial set of nodes has been patched, the user is asked to shut down the remaining nodes.
- After the local node is patched, the patch is propagated to the last set of nodes and the inventory is updated. The last instances are stopped on the remote nodes. You can then start the patched nodes (the first set of nodes) before patching the remaining nodes.

See Also:

- "Obtaining the Patch" on page 10-3
- "Preparing to Use OPatch" on page 10-3
- "Applying Patches" on page 10-6
- "Applying Patch Sets" on page 10-9
- "Troubleshooting Patch Deployment" on page 10-10

Applying Patch Sets

For instructions on applying the latest patch set to your Oracle RAC database and Oracle Clusterware installations on Oracle Linux, search for "Oracle 12*c* Release 1 (12.1) Support Status and Alerts" documentation on the My Oracle Support website.

This document provides a summary of the patch sets available for Oracle 12*c* Release 1. Using this document, you can easily locate and view the Patch Set Notes for your platform. The Oracle Database Patch Set Notes document contains the following information:

- System requirements and information about how to install or reinstall the patch set
- A list of all bugs fixed to date that are specific to Oracle Database for specified platform
- A list of known issues relating to Oracle Database for the specified platform

To locate the Patch Set notes on My Oracle Support:

1. Log in to My Oracle Support.
2. Select the **Patches & Updates** tab.
3. Select **Quick Links to the Latest Patchsets, Mini Packs, and Maintenance Packs**.
4. Under the heading Latest Oracle Server/Tools Patchsets, select **Oracle Database**.

   A list of operating systems appears.
5. Place your cursor over the entry that matches your operating system, or use the triangular arrows to search for your operating system.

   When you place the cursor over the entry for your operating system, for example, Linux x86, a list of database versions appears.
6. Select **12.1.0**

   The Advanced Search page appears.
7. Scroll to the bottom of this page to see the list of available patch sets.
8. Select the number in the Patch column for the patch set you want to view or download.

   The Patchset description and download page appears.
9. Click **View Readme** to see the patch set notes.

   On this page you can also click **Download** to download the patch to your computer.
10. If you choose to download the patch, then follow the instructions in the ReadMe file of the patch set to apply the patch set to your software.

See Also:

- "Preparing to Use OPatch"
- "Troubleshooting Patch Deployment"

Troubleshooting Patch Deployment

Applying patches to your Oracle RAC environment can be complicated at times. This section covers some common topics regarding troubleshooting patch deployment:

- Updating the Node List for OPatch
- About OPatch Log and Trace Files
- Resolving the "Not a valid patch area" Error
- Resolving the "Unable to remove a partially installed interim patch" Error

If you have problems applying a patch to your Oracle RAC database, then review these solutions to common problems. If the problem you encountered is not listed, then review the log and trace files and use My Oracle Support to obtain assistance.

See Also:

- *Oracle OPatch User's Guide for Windows and UNIX*
- "Preparing to Use OPatch" on page 10-3
- *Oracle Database 2 Day DBA* for more information about software management and patch releases
- "Viewing Oracle RAC Database Alert Log Messages" on page 8-40
- "About the Oracle Clusterware Alert Log" on page 8-32

Updating the Node List for OPatch

If OPatch does not automatically detect Oracle RAC or its nodes, then investigate the contents of the inventory and ensure they are complete.

To update the node list for OPatch:

If the list of nodes for your cluster is not complete, then you can update it by using Oracle Universal Installer and the `-updateNodeList` flag, as demonstrated in the following example:

```
Oracle_home/oui/bin/runInstaller -updateNodeList
ORACLE_ HOME=/u01/app/oracle/product/12.1.0/dbhome_1
CLUSTER_NODES=racnode1,racnode2,racnode3 -noClusterEnabled
```

See Also:

- "Troubleshooting Patch Deployment" on page 10-10
- *Oracle Universal Installer NextGen Installation Guide* for more information about updating the node list

About OPatch Log and Trace Files

Logging and tracing is a common aid for debugging. OPatch maintains logs for all `apply`, `rollback`, and `lsinventory` operations. The log files are located in the *`Oracle_home`*`/cfgtoollogs/opatch` directory. Each log file is tagged with the time stamp of the operation. Log files are named as *`opatch_mm-dd-yyyy_hh-mm-ss.log`*, where *`mm-dd-yyyy`* is the current date and *`hh-mm-ss`* is the current time. Each time you run OPatch, a new log file is created.

For example, if a log file is created on May 17, 2012 at 11:55 PM, then it is named as follows:

```
opatch_05-17-2012_23-55-00.log
```

OPatch also maintains an index of the commands processed by OPatch and the log files associated with it in the `opatch_history.txt` file located in the *`Oracle_home`*`/cfgtoollogs/opatch` directory. A sample of the `opatch_history.txt` file is as follows:

```
Date & Time : Tue Apr 26 23:00:55 PDT 2012
Oracle Home : /u01/app/oracle/product/12.1.0/dbhome_1/
OPatch Ver. : 12.1.0.0.0
Current Dir : /scratch/oui/OPatch
Command     : lsinventory
Log File    :
/u01/app/oracle/product/12.1.0/dbhome_1/cfgtoollogs/opatch/opatch-2012_Apr_26_
23-00-55-PDT_Tue.log
```

See Also:

- "Troubleshooting Patch Deployment" on page 10-10
- *Oracle OPatch User's Guide for Windows and UNIX*

Resolving the "Not a valid patch area" Error

You might get this error if the directory that the OPatch utility is using to do the patch does not match the template for what it is checking, or if the OPatch utility is run from an invalid directory.

The *`Patch_Shiphome`* directory should have the following structure:

- An `etc` subdirectory that has the metadata files
- A `files` subdirectory that has the patch files
- The `etc/config/inventory.xml` file and the `actions.xml` file under the same directory

To resolve the "Not a valid patch area" error:

- Perform one of the following actions:

  a. Remove the patch shiphome directory and re-create it with the proper structure (by extracting the files again).

  b. Start the OPatch utility from the directory where the patch to be installed has been unzipped and staged on disk.

  c. Use the following command when starting OPatch:

  ```
  opatch apply /Patch_Shiphome
  ```

  where *`Patch_Shiphome`* is the location where the patch has been staged on disk.

  See Also:

  - "Troubleshooting Patch Deployment" on page 10-10
  - *Oracle OPatch User's Guide for Windows and UNIX*

Resolving the "Unable to remove a partially installed interim patch" Error

If the patching process is interrupted, then you might get the error "Unable to remove a partially installed interim patch" when you try to install the patch a second time.

To resolve the partially installed patch error:

1. Ensure that the environment variable `ORACLE_HOME` is set to the Oracle home directory you are attempting to patch.
2. Go to the *`Oracle_home`*`/.patch_storage/`*`patch-id_timestamp`* directory and run the `restore.sh` script (or `restore.bat` on Windows platforms) as follows:

   ```
   Oracle_home/.patch_storage/patch-id_timestamp/restore.sh
   ```

3. On Linux and UNIX systems, use the *`Oracle_home`*`/.patch_storage/`*`patch-id_timestamp`*`/make.txt` file (if available) to modify your operating system environment, as follows:

```
/bin/sh make.txt
```

4. Attempt to apply the patch again.

See Also:

- "Troubleshooting Patch Deployment" on page 10-10
- *Oracle OPatch User's Guide for Windows and UNIX*

Upgrading the Oracle Software

An **upgrade** modifies most, if not all, of the installed Oracle software files, whereas a patch typically modifies only a few files. Upgrades are performed when you install a new software release or version, or apply a patch set. You can perform rolling upgrades, or you can shut down the Oracle Database and Oracle Clusterware software to perform the upgrade.

You perform out-of-place upgrades when upgrading the software to a new release. To perform an out-of-place upgrade, you install the Oracle Grid Infrastructure for a cluster into a new Grid home. When performing the upgrade, you specify the location of the new Grid home instead of selecting the existing software location.

When performing an out-of-place upgrade, the old and new version of the software are present on the nodes at the same time, each in a different home location, but only one version of the software is active at any given time. To upgrade your Oracle Database 11*g* software to Oracle Database 12*c* Release 1 (12.1), you install Oracle Grid Infrastructure for a cluster and Oracle Database into new Oracle home directories. At the end of the upgrade process, the active version of the software is changed to use the Oracle Grid Infrastructure for a cluster or Oracle Database 12*c* Release 1 (12.1) software.

You can use Database Upgrade Assistant (DBUA) to upgrade an existing database to the current release of Oracle Database. Database Upgrade Assistant (DBUA) guides you through the upgrade process and configures your database for the new release. DBUA automates the upgrade process and makes appropriate recommendations for configuration options such as tablespaces and online redo log files.

See Also:

- *Oracle Database 2 Day DBA* for more information about using DBUA to upgrade a database
- *Oracle Grid Infrastructure Installation Guide* for your operating system for more information on performing software upgrades
- *Oracle Database Upgrade Guide*

Glossary

administrator-managed database

An administrator-managed database is a database created on nodes that are not part of a server pool; the database or clusterware administrator manages these databases.

affinity

Any strategy that is expected to increase the probability that an application or client finds the requested data cached in the instance to which the request is routed.

affinity hint

A flag that indicates if affinity is active or inactive for a particular instance and service combination.

all node patching

A method of applying patches to the nodes in a cluster. When using the all node patching method, all the nodes that are part of the Oracle Real Application Clusters database are initially brought down and the patch is applied on all the nodes. After the patch is applied to all nodes, then the nodes are brought back up.

Application Continuity

A feature that can reestablish database sessions and resubmit pending work for recoverable errors, thereby protecting applications from transient failures. When a component fails, the recovery restores the state as it was before the failure. A user request is executed on time and at most once.

Automatic Workload Repository (AWR)

A built-in repository that exists in every Oracle Database. At regular intervals, Oracle Database makes a snapshot of all of its vital statistics and workload information and stores them in the AWR.

cache coherency

The synchronization of data in multiple caches so that reading a memory location through any cache returns the most recent data written to that location through any other cache. Sometimes called cache consistency.

Cache Fusion

A diskless cache coherency mechanism in Oracle Real Application Clusters that provides copies of blocks directly from a holding instance's memory cache to a requesting instance's memory cache.

cluster

Multiple interconnected computers or servers that appear as if they are one server to end users and applications.

cluster database

The generic term for an Oracle Real Application Clusters database.

cluster file system

A distributed file system that is a cluster of servers that collaborate to provide high performance service to their clients. Cluster file system software deals with distributing requests to storage cluster components.

Cluster Synchronization Services (CSS)

An Oracle Clusterware component that discovers and tracks the membership state of each node by providing a common view of membership across the cluster. CSS also monitors process health, specifically the health of the database instance. See also **OCSSD**.

Cluster Verification Utility (CVU)

A tool that verifies a wide range of Oracle RAC-specific components such as shared storage devices, networking configurations, system requirements, Oracle Clusterware, groups, and users.

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information. The network route provides, at a minimum, the location of the **listener** through use of a network address.

CRSD

A Linux or UNIX process that performs high availability recovery and management operations such as maintaining the OCR. Also manages application resources and runs as `root` user (or by a user in the `admin` group on Mac operating system X-based systems) and restarts automatically upon failure.

database services

Entities that you can define in Oracle RAC databases that enable you to group database workloads and route work to the optimal instances that are assigned to offer the service.

Dynamic Host Configuration Protocol (DHCP)

A network application protocol used by devices (DHCP clients) to obtain configuration information for operation in an Internet Protocol network. This protocol reduces system administration workload, enabling devices to be added to the network with little or no manual intervention.

Event Manager (EVM)

The background process that publishes Oracle Clusterware events. EVM scans the designated callout directory and runs all scripts in that directory when an event occurs.

Event Manager Daemon (EVMD)

A Linux or UNIX event manager daemon that starts the `racgevt` process to manage callouts.

Fast Application Notification (FAN)

Applications use FAN to enable rapid failure detection, balancing of connection pools after failures, and re-balancing of connection pools when failed components are repaired. The FAN notification process uses system events that Oracle publishes when cluster servers become unreachable or if network interfaces fail.

Fast Connection Failover

Fast Connection Failover provides high availability to FAN integrated clients, such as clients that use JDBC, OCI, or ODP.NET. If you configure the client to use fast connection failover, then the client automatically subscribes to FAN events and can react to database `UP` and `DOWN` events. In response, Oracle gives the client a connection to an active instance that provides the requested database service.

forced disk write

In Oracle Real Application Clusters, a particular data block can only be modified by one instance at a time. If one instance modifies a data block that another instance needs, then whether a forced disk write is required depends on the type of request submitted for the block.

Free pool

A default server pool used in policy-based cluster and capacity management of Oracle Clusterware resources. The free pool contains servers that are not assigned to any server pool.

General Parallel File System (GPFS)

General Parallel File System (GPFS) is a shared-disk IBM file system product that provides data access from all of the nodes in a homogenous or heterogeneous cluster.

Global Cache Service (GCS)

A process that implements Cache Fusion. It maintains the block mode for blocks in the global role. It is responsible for block transfers between instances. The Global Cache Service employs various background processes such as the Global Cache Service Processes (LMSn) and Global Enqueue Service Daemon (LMD).

Global Cache Service Processes (LMSn)

Processes that manage remote messages. Oracle RAC provides for up to 10 Global Cache Service Processes.

Global Cache Service (GCS) resources

Global resources that coordinate access to data blocks in the buffer caches of multiple Oracle RAC instances to provide cache coherency.

global database name

The full name of the database that uniquely identifies it from any other database. The global database name is of the form *database_name.database_domain*—for example: TEST.US.EXAMPLE.COM

global dynamic performance views

Dynamic performance views storing information about all open instances in an Oracle Real Application Clusters cluster. (Not only the local instance.) In contrast, standard dynamic performance views (V$) only store information about the local instance.

Global Enqueue Service (GES)

A service that coordinates enqueues that are shared globally.

Global Enqueue Service Daemon (LMD)

The resource agent process that manages requests for resources to control access to blocks. The LMD process also handles deadlock detection and remote resource requests. Remote resource requests are requests originating from another instance.

Global Enqueue Service Monitor (LMON)

The background LMON process monitors the entire cluster to manage global resources. LMON manages instance deaths and the associated recovery for any failed instance. In particular, LMON handles the part of recovery associated with global resources. LMON-provided services are also known as Cluster Group Services.

Global Services Daemon (GSD)

A component that receives requests from SRVCTL to execute administrative job tasks, such as startup or shutdown. The command is executed locally on each node, and the results are returned to SRVCTL. GSD is installed on the nodes by default.

Grid home

The Oracle Home directory for Oracle Grid Infrastructure for a cluster, which includes Oracle Clusterware and Oracle ASM.

grid infrastructure

The software that provides the infrastructure for an enterprise grid architecture. Oracle Grid Infrastructure for a cluster bundles these infrastructure software products into one product. In an Oracle cluster, Oracle Grid Infrastructure for a cluster includes Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM). For a standalone Oracle Database server, Oracle Grid Infrastructure for a cluster includes Oracle Restart and Oracle ASM.

Grid Naming Service (GNS)

A generic service which resolves the names of hosts in a delegated normal DNS zone by mapping them to IP addresses within the zone. GNS enables the use of **Dynamic Host Configuration Protocol (DHCP)** address for Oracle RAC database nodes, simplifying deployment. GNS also resolves host names passed back from a **SCAN** listener.

high availability

Systems with redundant components that provide consistent and uninterrupted service, even following hardware or software failures. This involves some degree of redundancy.

High Availability Cluster Multi-Processing (HACMP)

High Availability Cluster Multi-Processing is an IBM AIX-based high availability cluster software product. HACMP has two major components: high availability (HA) and cluster multi-processing (CMP).

instance

For an Oracle RAC database, each node in a cluster usually has one instance of the running Oracle software that references the database. When a database is started, Oracle allocates a memory area called the System Global Area (SGA) and starts one or more Oracle processes. This combination of the SGA and the Oracle processes is called an instance. Each instance has unique Oracle System Identifier (SID), instance name, rollback segments, and thread ID.

instance membership recovery

The method used by Oracle RAC guaranteeing that all cluster members are functional or active. IMR polls and arbitrates the membership. Any members that do not show a heartbeat by way of the control file or who do not respond to periodic activity inquiry messages are presumed terminated.

instance name

Represents the name of the instance and uniquely identifies a specific instance when clusters share common services names. The `INSTANCE_NAME` parameter in the instance initialization file, `init`*`sid`*`.ora`, identifies the instance name. The instance name equals the Oracle System Identifier (SID).

instance number

A number that associates extents of data blocks with particular instances. The instance number enables you to start an instance and ensure that it uses the extents allocated to it for inserts and updates. This ensures that an instance does not use space allocated for other instances.

interconnect

The private network communication link used to synchronize the memory cache of the nodes in the cluster.

listener

A process that resides on the server and whose responsibility is to listen for incoming client connection requests and manage the traffic to the server. When a client requests a network session with a database server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the database server.

logical transaction ID

A globally unique identifier that defines a transaction from the application perspective.

Logical Volume Manager (LVM)

A generic term that describes Linux or UNIX subsystems for online disk storage management.

Inter-Process Communication (IPC)

A high-speed operating system-dependent transport component. The IPC transfers messages between instances on different nodes. Also referred to as the interconnect.

Master Boot Record (MBR)

A program that executes when a computer starts. Typically, the MBR resides on the first sector of a local hard disk. The program begins the startup process by examining the partition table to determine which partition to use for starting the computer. The

MBR program then transfers control to the boot sector of the startup partition, which continues the startup process.

minimum downtime patching

In minimum downtime patching, the nodes are divided into two sets. The first set is shut down and the patch is applied to it. The second set is then shut down. The first set is brought up and then the patch is applied to the second set. After the patch is applied to the second set, those nodes are also brought up, finishing the patching operation.

multicast Domain Name Server (mDNS)

A part of Zero Configuration Networking (Zeroconf), mDNS provides the ability to address hosts using DNS-like names without the need of an existing, managed DNS server.

mutables

Data returned from **mutable objects**. An example of a mutable is a call to `SYSTIMESTAMP` function.

mutable objects

A function that obtains a new value every time it is called. Examples of mutable objects are `SYSDATE`, `SYSTIMESTAMP`, `SYS_GUID`, and *`sequence`*`.NEXTVAL`.

net service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a net service name in a connect string for the database service to which they want to connect.

Network Attached Storage (NAS)

Storage that is attached to a server by way of a network.

Network Interface Card (NIC)

A card that you insert into a computer to connect the computer to a network.

network switch

A hardware device that connects computers within a network.

Network Time Protocol (NTP)

An Internet standard protocol, built on top of TCP/IP, that ensures the accurate synchronization to the millisecond of the computer clock times in a network of computers.

node

A node is a computer on which the Oracle Clusterware software is installed or will be installed.

Object Link Manager (OLM)

The Oracle interface that maps symbolic links to logical drives and displays them in the OLM graphical user interface.

OCSSD

A Linux or UNIX process that manages the Cluster Synchronization Services (CSS) daemon. Manages cluster node membership and runs as `oracle` user; failure of this process results in cluster restart.

optimal flexible architecture (OFA)

A set of file naming and configuration guidelines created to ensure reliable Oracle installations that require little maintenance.

Oracle Base directory

The mountpoint for all software installations performed by a particular user. An Oracle base directory can contain multiple Oracle homes for Oracle software products, either of the same or different releases, all installed by the same operating system user. The Oracle Base directory is also the directory where the software parameter files, log files, trace files, and so on, associated with a specific installation owner are located.

Oracle Cluster File System (OCFS)

The Oracle proprietary cluster file system software that is available for Linux platforms.

Oracle Cluster Registry (OCR)

The Oracle RAC configuration information repository that manages information about the cluster node list and instance-to-node mapping information. The OCR also manages information about Oracle Clusterware resource profiles for customized applications.

Oracle Clusterware

This is clusterware that is provided by Oracle to manage cluster database processing including node membership, group services, global resource management, and high availability functions.

Oracle Home directory

The binary location for a particular software installation.

Typically The Oracle Home directory is a subdirectory of the Oracle Base directory for the software installation owner. However, with Oracle Grid Infrastructure for a cluster, the Oracle Home directory (in this case, the Grid home) is located outside of the Oracle Base directory for the Oracle Grid Infrastructure for a cluster installation owner, because the path of the Grid home is changed to `root` ownership.

Oracle Interface Configuration Tool (OIFCFG)

A command-line tool for both single-instance Oracle databases and Oracle RAC databases that enables you to allocate and de-allocate network interfaces to components, direct components to use specific network interfaces, and retrieve component configuration information. The Oracle Universal Installer (OUI) also uses OIFCFG to identify and display available interfaces.

Oracle Inventory directory

The Oracle Inventory directory is the central inventory location for all Oracle software installed on a server.

Oracle Notification Services (ONS)

A publish and subscribe service for communicating information about all FAN events.

Oracle Universal Installer (OUI)

A tool to install Oracle Clusterware, the Oracle relational database software, and the Oracle Real Application Clusters software. You can also use the Oracle Universal Installer to launch the Database Configuration Assistant (DBCA).

policy-managed database

A policy-managed database is created using a server pool. Oracle Clusterware allocates and reassigns capacity based on policies you define, enabling faster resource failover and dynamic capacity assignment.

recoverable error

A class of errors that arise because of an external system failure, independently of the application session logic that is executing. Recoverable errors occur following planned and unplanned outages of networks, nodes, storage, and databases. An example of a nonrecoverable error is submission of invalid data values.

Recovery Manager (RMAN)

An Oracle tool that enables you to back up, copy, restore, and recover data files, control files, and archived redo logs. It is included with the Oracle server and does not require separate installation. You can invoke RMAN as a command line utility from the operating system (O/S) prompt or use the GUI-based Enterprise Manager Backup Manager.

redo thread

The redo generated by a database **instance**.

rolling patching

In Rolling Patching, one node (or group of nodes) is shutdown, the patch applied and the node brought back up again. This is repeated for each node in the cluster until all the nodes in the Real Application Clusters are patched.

Run-time Connection Load Balancing

Enables Oracle to make intelligent service connection decisions based on the connection pool that provides the optimal service for the requested application based on current workloads. The JDBC, ODP.NET, and OCI clients are integrated with the load balancing advisory; you can use any of these client environments to provide run-time connection load balancing.

scalability

The ability to add additional nodes to Oracle Real Application Clusters applications and achieve markedly improved scale-up and speed-up.

SCAN

A single name, or network alias, for the cluster. Oracle Database 11*g* database clients use SCAN to connect to the database. SCAN can resolve to multiple IP addresses, reflecting multiple listeners in the cluster handling public client connections.

Secure Shell (SSH)

A program for logging into a remote computer over a network. You can use SSH to execute commands on a remote computer and to move files from one computer to another. SSH uses strong authentication and secure communications over insecure channels.

Server Control (SRVCTL) Utility

Server Management (SRVM) comprises the components required to operate Oracle Enterprise Manager in Oracle Real Application Clusters. The SRVM components, such as the Intelligent Agent, Global Services Daemon, and SRVCTL, enable you to manage cluster databases running in heterogeneous environments through an open client/server architecture using Oracle Enterprise Manager.

server pool

A server pool is a logical division of nodes in a cluster into a group to support policy-managed databases.

shared everything

A database architecture in which all instances share access to all of the data.

singleton services

Services that run on only one instance at any one time.

split brain syndrome

Where two or more instances attempt to control a cluster database. In a two-node environment, for example, one instance attempts to manage updates simultaneously while the other instance attempts to manage updates.

system identifier (SID)

The Oracle system identifier (SID) identifies a specific instance of the running Oracle software. For an Oracle Real Application Clusters database, each node within the cluster has an instance referencing the database.

thread

Each Oracle instance has its own set of online redo log groups. These groups are called a thread of online redo. In non-Oracle Real Application Clusters environments, each database has only one thread that belongs to the instance accessing it. In Oracle Real Application Clusters environments, each instance has a separate thread, that is, each instance has its own online redo log. Each thread has its own current log member.

thread number

An identifier for the redo thread to be used by an instance, specified by the `INSTANCE_NUMBER` initialization parameter. You can use any available redo thread number but an instance cannot use the same redo thread number as another instance.

transaction

A logical unit of work that contains one or more SQL statements. All statements in a transaction commit or roll back together. The use of transactions is one of the most important ways that a database management system differs from a file system.

Transaction Guard

A database feature that uses a **logical transaction ID** to prevent the possibility of a client application submitting duplicate transactions after a **recoverable error**.

transaction ID

An identifier that is unique to a transaction and represents the undo segment number, slot, and sequence number.

transaction idempotence

The ability to return a guaranteed outcome for a **transaction**: whether it committed and whether the call was completed.

transparent application failover (TAF)

A run-time failover for high-availability environments, such as Oracle Real Application Clusters, TAF refers to the failover and re-establishment of application-to-service connections. It enables client applications to automatically reconnect to the database if the connection fails, and optionally resume a SELECT statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI) library.

voting disk

A file that manages information about node membership.

Index