Oracle® Database

2 Day + Real Application Clusters Guide

10*g* Release 2 (10.2)

B28759-01

December 2006

ORACLE®

Oracle Database 2 Day + Real Application Clusters Guide, 10*g* Release 2 (10.2)

B28759-01

Copyright © 2006, Oracle. All rights reserved.

Primary Authors: Mark Bauer, Janet Stern, Douglas Williams

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface xi

Audience xi
Documentation Accessibility xi
Related Documents xii
Conventions xii

1 Introduction to Oracle Database 2 Day + Real Application Clusters Guide

About This Guide 1-1
What This Guide Is Not 1-1
Related Materials 1-1
Oracle Real Application Clusters Documentation Overview 1-2
Goal of This Guide 1-2
Installing Oracle RAC on Different Operating Systems 1-2
Useful Oracle RAC Guides 1-2
About Oracle Clusterware and Oracle Real Application Clusters 1-3
About Oracle Automatic Storage Management 1-3
Tools for Installing, Configuring and Managing Oracle RAC 1-4

2 Preparing Your Cluster

Checking Requirements 2-1
Checking the Hardware Requirements 2-1
Identifying Network Requirements 2-3
Verifying the Installed Operating System and Software Requirements 2-4
Preparing the Server 2-5
Configuring Operating System Users and Groups 2-5
Configuring the Secure Shell 2-6
Generating RSA and DSA Keys 2-7
Adding the Keys to an Authorized Key File 2-8
Configuring SSH User Equivalency 2-9
Configuring the Operating System Environment 2-10
Configuring the Network 2-11
Verifying the Network Configuration 2-12
Preparing the Operating System and Software 2-14
Setting the Time on Both Nodes 2-14
Configuring Kernel Parameters 2-15

Performing Platform-Specific Configuration Tasks 2-15
Configuring Installation Directories and Shared Storage 2-15
Deciding on a Shared Storage Solution 2-15
Configuring the Raw Storage Devices and Partitions 2-16
Configuring Raw Devices on Red Hat Enterprise Linux 4.0 2-19
Choosing an Oracle Base Directory 2-20
Choosing an Oracle Clusterware Home Directory 2-20

3 Installing and Configuring Oracle Clusterware and Oracle RAC

Preparing the Oracle Media Installation File 3-1
Installing Oracle Clusterware 10*g* 3-2
Configuring the Operating System Environment 3-2
Verifying the Configuration Using the Cluster Verification Utility 3-3
Using Oracle Universal Installer to Install Oracle Clusterware 3-3
Completing the Oracle Clusterware Configuration 3-10
Configuring Automatic Storage Management in an ASM Home Directory 3-11
Verifying Your ASM Installation 3-16
Installing the Oracle Database Software and Creating a Cluster Database 3-16
Verifying your Oracle RAC Database Installation 3-23
Configuring the Operating System Environment for Database Management 3-23
Update the oratab File 3-23
Reconfigure the User Shell Profile 3-24
Performing Postinstallation Tasks 3-25
Verifying the Clusterware Installation 3-25
Backing Up the Voting Disk 3-25
Downloading and Installing RDBMS Patches 3-25
Verifying Oracle Enterprise Manager Operations 3-26
Recommended Postinstallation Tasks 3-26
Backing Up the root.sh Script 3-26
Configuring User Accounts 3-26
Converting an Oracle Database to an Oracle RAC Database 3-27
Checking the Prerequisites 3-27
Making a Preconfigured Copy of the Single-Instance Database 3-27
Performing the Preinstallation Steps 3-28
Validating the Cluster 3-28
Copying the Preconfigured Database Files 3-28
Installing the Oracle Database 10*g* Software with Real Application Clusters 3-28

4 Administering Database Instances and Cluster Databases

Overview of Oracle RAC Database Management 4-1
Administering Oracle RAC with Enterprise Manager 4-2
Overview of Oracle Real Application Clusters: Oracle By Example Series 4-3
Starting and Stopping Oracle RAC Databases and Database Instances 4-3
Viewing and Modifying Oracle RAC Initialization Parameters 4-4
Configuring Initialization Parameters for an Oracle RAC Database 4-4
Parameters that Must Have Identical Settings on All Instances 4-4
Parameters that Must Have Unique Settings on All Instances 4-5

Parameters that Should Have Identical Settings on All Instances 4-5
Editing Initialization Parameter Settings for an Oracle RAC Database 4-6
Modifying Initialization Parameter for Oracle RAC Using the Current Tab 4-6
Modifying Initialization Parameter for Oracle RAC Using the SPFile Tab 4-7
Example: Modifying the OPEN_CURSORS Parameter 4-7
Modifying the SERVICE_NAMES Parameter for Oracle RAC 4-8
Configuring the Server Parameter File for Oracle Real Application Clusters 4-8
Administering Storage in Oracle RAC 4-8
Administering Automatic Undo Management in Oracle RAC 4-8
Administering Automatic Storage Management in Oracle RAC 4-9
About Automatic Storage Management Components in Oracle RAC 4-9
Modifying Disk Group Configurations for ASM in Oracle RAC 4-9
Standalone ASM Disk Group Management 4-9
Administering ASM Instances and Disk Groups with Enterprise Manager 4-10
Administering Redo Logs in Oracle RAC 4-10
About Redo Log Groups and Redo Threads in Oracle RAC Databases 4-10
Using Enterprise Manager to View and Create Online Redo Log Files 4-11
Exploring Your Cluster Database: Oracle By Example Series 4-11

5 Administering Oracle Clusterware Components

About Oracle Clusterware 5-1
Backing Up and Recovering Voting Disks 5-2
Backing Up Voting Disks 5-2
Recovering Voting Disks 5-2
Adding and Removing Voting Disks 5-3
Backing Up and Recovering the Oracle Cluster Registry 5-3
Viewing Available OCR Backups 5-3
Backing Up the OCR 5-3
Recovering the OCR 5-4
Restoring the Oracle Cluster Registry from Automatically Generated OCR Backups 5-4
Recovering the OCR from an OCR Export File 5-5
Changing the Oracle Cluster Registry Configuration 5-6
Adding an OCR Location 5-6
Replacing an OCR 5-6
Repairing an Oracle Cluster Registry Configuration on a Local Node 5-7
Removing an Oracle Cluster Registry 5-7
Troubleshooting the Oracle Cluster Registry 5-8
Using the OCRCHECK Utility 5-8
Resolving Common Oracle Cluster Registry Problems 5-9

6 Administering Backup and Recovery

Overview of Oracle RAC Database Backup and Recovery 6-1
Deploying a Flash Recovery Area in Oracle Real Application Clusters 6-2
Archiving in Oracle Real Application Clusters 6-3
Configuring Archiving for Your Oracle RAC Database 6-3
About Instance Access to Archived Log Files 6-3

Credentials for Performing Enterprise Manager Backup and Recovery 6-3
Configuring Backup Settings 6-4
Performing Backups of Your Oracle RAC Database 6-4
About Parallelism and Backups Across Multiple Channels 6-5
Backing Up Archived Logs 6-5
Preparing to Restore and Recover Your Oracle RAC Database 6-5
Configuring Access to the Redo Log Files 6-6
Putting the Oracle RAC Database Instances into the Correct State 6-6
Recovering Your Oracle RAC Database 6-6
Recovering the Parameter File from an Automatic File Backup 6-7
About Restoring Archived Log Files 6-7
Performing Recovery Using Parallelism 6-7
Managing Your Database Backups 6-8
Displaying Backup Reports for Your Oracle RAC Database 6-8
Performing Backups and Recovering Your Database: Oracle By Example Series 6-9

7 Managing Database Workload Using Services

About Workload Management 7-1
About Oracle Services 7-1
Configuring Services for High Availability 7-2
About the Database Resource Manager 7-2
About Oracle RAC High Availability Framework 7-3
Enabling the Load Balancing Advisory 7-3
About Connection Load Balancing 7-4
Client-Side Load Balancing 7-5
Server-Side Load Balancing 7-5
About Runtime Connection Load Balancing 7-6
About Fast Application Notification (FAN) 7-7
About FAN Callouts 7-7
Creating Services 7-8
Configuring Oracle Net to Support Services 7-10
Administering Services 7-11
Administering Services with Enterprise Manager 7-12
Using the Cluster Managed Database Services Page 7-12
Using the Cluster Managed Database Services Detail Page 7-12
Configuring Service-Level Thresholds 7-13
Managing Services Using the Database Configuration Assistant 7-14
Transparent Application Failover: Oracle By Example Series 7-15

8 Monitoring and Troubleshooting

Using Performance Views in Oracle Real Application Clusters 8-1
GV$ Views 8-1
Monitoring Oracle RAC Database Performance 8-2
Using the Cluster Database Performance Page 8-2
Reviewing the Chart for Cluster Host Load Average 8-3
Reviewing the Chart for Global Cache Block Access Latency 8-4
Reviewing the Cluster Cache Coherency Page 8-4

Reviewing the Chart for Average Active Sessions 8-6
Reviewing the Charts for Database Throughput 8-6
Reviewing Other Performance-Related Charts 8-7
Top Consumers Page 8-7
Top Sessions Page 8-7
Instance Activity Page 8-7
Top Segments Page 8-8
Database Locks Page 8-8
Using the Automatic Workload Repository in Oracle RAC Environments 8-8
Troubleshooting Configuration Problems in Oracle RAC Environments 8-8
Using CRSCTL to Diagnose Cluster Issues 8-9
Obtaining Oracle Clusterware Component Names 8-9
Enabling Debugging of Oracle Clusterware Components 8-9
Enabling and Disabling Oracle Clusterware Daemons 8-10
Locating the Oracle Clusterware Alert Log 8-10
Enabling Debugging for an Oracle Clusterware Resource 8-10
Checking the Status of the Oracle Clusterware Installation 8-10
Locating the Oracle Clusterware Component Log Files 8-11
Running the Oracle Clusterware Diagnostics Collection Script 8-11
Using the Cluster Verification Utility to Diagnose Problems 8-11
Enabling Tracing 8-12
Checking the Settings for the Interconnect 8-12
Troubleshooting a Node with Status of UNKNOWN 8-13
Verifying the Existence of Node Applications 8-13
Verifying the Integrity of Oracle Clusterware Components 8-13
Verifying the Integrity of the Oracle Cluster Registry 8-13
Verifying the Integrity of Your Entire Cluster 8-13
Viewing Oracle RAC Database Alerts 8-14
Viewing Oracle RAC Database Alert Log Messages 8-14

9 Adding Nodes and Instances

Preparing Access to the New Node 9-1
Extending the Oracle Clusterware Home Directory 9-2
Extending the Oracle Automatic Storage Management Home Directory 9-3
Extending the Oracle RAC Software Home Directory 9-3
Creating a Listener on the New Node 9-4
Adding a New Cluster Instance on the New Node 9-5

10 Managing Oracle Software and Applying Patches

Configuring the Enterprise Manager Patch Interface 10-1
Setting Oracle Meta*Link* Credentials 10-1
Running the Refresh_From_Metalink Job 10-2
Obtaining the Patch 10-2
Preparing to Use OPatch 10-4
Check ORACLE_HOME Environment Variable 10-4
Perform a Backup 10-4

Stage the Patch on Each Node 10-4
Update the PATH Environment Variable 10-4
Configure SSH User Equivalency 10-5
Applying Patches 10-5
All Node Patching 10-6
Rolling Patching 10-8
Minimum Downtime Patching 10-11
Applying Patch Sets 10-12
Troubleshooting Patch Deployment 10-13
Updating the Node List for OPatch 10-13
Viewing Log and Trace Files 10-13
Resolving the "Not a valid patch area" Error 10-14
Resolving the "Unable to remove a partially installed interim patch" Error 10-14

Index

List of Tables

5–1 Common OCR Problems and Solutions 5-9
8–1 Cluster Cache Coherency Charts 8-5
10–1 Patching Oracle Home Directories 10-6

Preface

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

> **Note:** For Linux operating systems other then Red Hat Linux, see *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide for Linux*. 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

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading

technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see the following in the Oracle Database Release 10.2 documentation set:

- *Oracle Real Application Clusters Installation and Configuration Guide*
- *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide for Linux*
- *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Database 2 Day DBA*

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.

1

Introduction to Oracle Database 2 Day + Real Application Clusters Guide

This chapter provides an overview of Oracle Real Application Clusters (Oracle RAC) environments. This chapter includes the following sections:

- About This Guide
- About Oracle Clusterware and Oracle Real Application Clusters
- Tools for Installing, Configuring and Managing Oracle RAC

About This Guide

This is an Oracle RAC database administration, task-oriented guide that shows you how to configure and manage the environment for Oracle Clusterware and Oracle RAC. This guide also explains how to create an Oracle RAC database and how to perform routine Oracle RAC database administrative tasks.

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 Red Hat Linux, but you do not need to be a Linux expert to use this guide.

What This Guide Is Not

This guide is not a comprehensive description about Oracle RAC. It describes concepts only when necessary for completing a particular task.

See Also:

- *Oracle Database Concepts* for more information about Oracle Database concepts
- *Oracle Database Administrator's Guide* for more information about Oracle Database administrative tasks

Related Materials

This 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, which is available on the Web, and Oracle University instructor-led classes.

Some of the chapters in this guide have an associated OBE lesson. The OBE lesson guides you through some of the tasks in the chapter or related tasks, and includes

annotated screen shots. In some cases, the OBE lesson provides additional information to help you complete a task.

At the end of a chapter, you might find a link to that chapter's associated OBE lesson. The home page for the 2 Day + Real Application Clusters Oracle By Example series is

http://www.oracle.com/technology/obe/10gr2_2day_dba/index.html

Oracle Real Application Clusters Documentation Overview

This guide describes how to install, configure, and manage Oracle RAC and provides examples for how you could do this on a two-node cluster. This guide is for DBAs who have experience with single-instance Oracle environments and have read *Oracle Database 2 Day DBA*.

Goal of This Guide

The goal of this guide is to show you how to install and deploy Oracle Clusterware and Oracle RAC on a two-node cluster using the Red Hat Linux operating system. To do this, you may need to work with your system administrator.

Installing Oracle RAC on Different Operating Systems

If you plan to install and configure Oracle RAC on an operating system other than Red Hat Linux, 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 Real Application Clusters on a different platform or different operating system version than Red Hat Linux, you need to refer to the installation and configuration guide for that platform. For example, if you are installing Oracle Real Application Clusters on the Solaris operating system, then you would use the following guide:

Oracle Database Oracle Clusterware and 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 Red Hat Linux and another node in the same cluster running Solaris UNIX. All nodes must run the same operating system, that is, they must be binary compatible. In an active data-sharing environment, like Oracle RAC, Oracle does not support machines having different chip architectures. However, you can have machines of different speeds and size in the same cluster.

Useful Oracle RAC Guides

The following is a list of other useful Oracle RAC or related documentation:

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

> **Note:** Additional information for this release may be available in the Oracle Database 10*g* Release 2 (10.2) README or Release Notes. You can locate these documents on your Oracle product installation media.

About Oracle Clusterware and Oracle Real Application Clusters

Oracle RAC extends Oracle Database so that you can store, update, and efficiently retrieve data from multiple instances on different servers at the same time. Oracle RAC provides the software that allows the servers to work together in what is called a cluster. The physical structures that make up 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.

Single-instance Oracle databases have a one-to-one relationship between datafiles and the instance. Oracle RAC environments, however, have a one-to-many relationship between datafiles and instances. In Oracle RAC environments, multiple cluster database instances form a single database. The instances can be on different servers, referred to as *hosts* or nodes. The combined processing power of the multiple servers provides greater throughput and scalability than is available from a single server.

Each cluster database instance in an Oracle RAC cluster uses its own memory structures and background processes. Oracle RAC uses Cache Fusion to synchronize the data stored in the buffer cache of each cluster 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 a data block located in the buffer cache of one instance is required by 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.

The Oracle RAC infrastructure is also a key component for implementing the Oracle enterprise grid computing architecture. Having multiple cluster database instances accessing a single database prevents the server from being a single point of failure. Any packaged or custom application that ran well on a single-instance Oracle database will perform well on an Oracle RAC database without requiring code changes.

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

About Oracle Automatic Storage Management

With Oracle RAC, each instance must have access to the datafiles and recovery files for the Oracle RAC database. Using Automatic Storage Management (ASM) is an easy way to satisfy this requirement.

ASM is an integrated, high-performance database file system and disk manager. ASM is based on the principle that the database should manage storage instead of requiring an administrator to do it. ASM eliminates the need for you to directly manage potentially thousands of Oracle database files.

ASM groups the disks in your storage system into one or more disk groups. You manage a small set of disk groups and ASM automates the placement of the database files within those disk groups.

ASM provides the following benefits:

- **Striping**—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**—ASM can increase data availability by optionally mirroring any file. 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, ASM can continue to access affected files by accessing mirrored copies on the surviving disks in the disk group.

- **Online storage reconfiguration and dynamic rebalancing**—ASM permits you to add or remove disks from your disk storage system while the database is operating. When you add a disk, 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**—ASM further reduces administration tasks by enabling files stored in ASM disk groups to be managed by Oracle Database. ASM automatically assigns file names when files are created, and automatically deletes files when they are no longer needed.

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

Oracle recommends that you use ASM for your database file storage, instead of raw devices or the operating system file system. However, databases can have a mixture of ASM files and non-ASM files.

Tools for Installing, Configuring and Managing Oracle RAC

The following is a description of the tools used for installing, configuring, and managing an Oracle RAC database:

- Oracle Universal Installer (OUI)–OUI installs the Oracle Clusterware and the Oracle Database software with Oracle RAC. After you configure the nodes that you want to use in your cluster, OUI installs the Oracle software on the specified nodes using a network connection.
- Cluster Verification Utility (CVU)–The CVU is a command-line tool that you can use to verify a range of cluster and Oracle RAC components such as shared storage devices, networking configurations, system requirements, and Oracle Clusterware, as well as operating system groups and users. You can use the CVU for preinstallation as well as postinstallation checks of your cluster environment. The CVU is especially useful during preinstallation and during installation of Oracle Clusterware and Oracle RAC components. OUI runs the CVU after the Oracle Clusterware installation to verify your environment.
- Oracle Enterprise Manager –Oracle Enterprise Manager has both the Database Control and Grid Control graphical user interfaces (GUIs) for managing single-instance and Oracle RAC environments.
- Server Control (SRVCTL)–SRVCTL is a command-line interface that you can use to manage an Oracle RAC database from a single point. You can use SRVCTL to start and stop the database and instances, and to delete or move instances and services. You can also use SRVCTL to add services and manage configuration information. You use SVRCTL to start and stop a group of applications that

includes virtual IP addresses, Listeners, Oracle Notification Services, node-level applications, and Oracle Enterprise Manager agents (for maintenance purposes).

- Cluster Ready Services Control (CRSCTL)–CRSCTL is a command-line tool that you can use to manage Oracle Clusterware. You can use CRSCTL to start and stop Oracle Clusterware and to determine the current status of your Oracle Clusterware installation.

2

Preparing Your Cluster

This chapter contains the information that your system administrator and network administrator need to help you configure the two nodes in your cluster. This chapter assumes a basic understanding of the Red Hat Linux operating system. In some cases, you may need to refer to details in *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide for Linux*. In addition, you must have `root` privileges to perform the tasks in this chapter.

This chapter includes the following sections:

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

Checking Requirements

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

- Checking the Hardware Requirements
- Identifying Network Requirements
- Verifying the Installed Operating System and Software Requirements

Checking the Hardware Requirements

Each node that you want to make part of your Oracle Clusterware, or Oracle Clusterware and Oracle RAC installation, must satisfy the minimum hardware requirements of the software. These hardware requirements can be categorized as follows:

- Physical memory (at least 1 gigabyte (GB) of RAM)
- Swap space (at least 2 GB of available swap space)
- Temporary space (at least 400 megabytes (MB))
- Processor type (CPU) that is certified with the version of the Oracle software being installed

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

You will need at least 1.5 GB of available disk space for the Oracle Database home directory and 1.5 GB of available disk space for the Oracle Automatic Storage Management (ASM) home directory. You will also need 120 MB of disk available space for the Oracle Clusterware software installation. For best performance and protection, you should have multiple disks, each using a different disk controller.

An Oracle RAC database is a *shared everything* database. All datafiles, control files, redo log files, and the server parameter file (SPFILE) in Oracle RAC environments must reside on shared storage that is accessible by all the instances in the cluster database. The Oracle RAC installation that is described in this guide uses Oracle ASM for the shared storage of the 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 more than one voting disk, 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 as well as 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:** Both the voting disks and the OCR must reside on shared devices that you configure before you install Oracle Clusterware and Oracle RAC.

These Oracle Clusterware components require the following additional disk space:

- Two Oracle Clusterware Registry files, 256 MB each, or 512 MB total disk space
- Three voting disk files, 256 MB each, or 768 MB total disk space

For voting disk file placement, ensure that each voting disk is configured so that it does not share any hardware device or disk, or other single point of failure. See "Configuring the Raw Storage Devices and Partitions" on page 2-16 for more information about configuring Oracle Clusterware files.

> **See Also:** *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform for information about exact requirements

Identifying Network Requirements

An Oracle RAC cluster comprises two or more nodes that are linked by a private interconnect. The interconnect serves as the communication path between nodes in the cluster. Each cluster database instance uses the interconnect for messaging to synchronize each instance's use of the shared resources. Oracle RAC also uses the interconnect to transmit data blocks that are shared between the instances.

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 used by Oracle RAC is the same interconnect that Oracle Clusterware uses. This interconnect should be a private interconnect, meaning it is *not* be accessible by nodes that are not members of the cluster.

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

- Each node needs at least two network interface cards, or **network adapters**. One adapter is for the public network and the other adapter is for the private network used by the interconnect. You should install additional network adapters on a node if that node:
  - 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.

    Note: For the most current information about supported network protocols and hardware for Oracle RAC installations, refer to the Certify pages on OracleMeta*Link*, which is located at

    `http://metalink.oracle.com`

- You must have at least three IP addresses available for each node:
  1. An IP address with an associated host name (or network name) for the public interface.
  2. A private IP address with a host name for each private interface.

     Note: Oracle recommends that you use private network IP addresses for the private interfaces (for example: 10.*.*.* or 192.168.*.*).

  3. One virtual IP address with an associated network name. Select a virtual IP (VIP) address that meets the following requirements:
     - The VIP address and associated network name are currently unused.
     - The VIP is on the same subnet as your public interface.
- Public interface names must be the same for all nodes. If the public interface on one node uses the network adapter `eth0`, then you must configure `eth0` as the public interface on *all* nodes.
- You should configure the same private interface names for all nodes as well. If `eth1` is the private interface name for the first node, then `eth1` should be the private interface name for your second node.

- For the private network, the end points of all designated interconnect interfaces must be completely reachable on the network. There should be no node that is not accessible by other nodes in the cluster using the private network.

To determine what interfaces are configured on a node running Red Hat Linux, use the following command as the `root` user:

```
# /sbin/ifconfig
```

You may need to work with your system or network administrator to obtain IP addresses for each node. See "Configuring the Network" on page 2-11 for more information about configuring the IP addresses and interface names.

Verifying the Installed Operating System and Software Requirements

Refer to *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform for information about exact requirements. These requirements can include any of the following:

- The operating system version
- The kernel version of the operating system
- 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 version that is not supported by Oracle Database 10*g* Release 2 (10.2), then you must first upgrade your operating system before installing Oracle Real Application Clusters 10*g*.

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

1. To determine which distribution and version of Linux is installed, run the following command as the `root` user:

   ```
   cat /etc/issue
   ```

2. 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*. To determine if the required errata level is installed, use the following procedure as the `root` user:

   ```
   uname -r
   2.4.21-27.EL
   ```

   The output in the previous example shows that the kernel version is 2.4.21, and the errata level (EL) is 27. 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.

3. To ensure there are no operating system issues effecting installation, make sure you have installed all the operating system patch updates and packages that are listed in *Oracle Clusterware and Oracle Real Application Clusters Installation Guide* for your platform. If you are using RedHat Linux, 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 Web site.

Preparing the Server

In this section, you will perform the following tasks:

- Configuring Operating System Users and Groups
- Configuring the Secure Shell
- Configuring SSH User Equivalency
- Configuring the Operating System Environment

Configuring Operating System Users and Groups

Depending on whether or not this is the first time Oracle software is being installed on this system, you may need to create operating system groups.

The following operating system groups are required if you are installing Oracle RAC:

- The OSDBA group (typically, `dba`)
- The Oracle Inventory group (typically, `oinstall`)

The following operating system users are required for all installations:

- A user that owns the Oracle software (typically, `oracle`)
- An unprivileged user (for example, the `nobody` user on Linux systems)

A single Oracle Inventory group is required for all installations of Oracle software on the system. After the first installation of Oracle software, you must use the same Oracle Inventory group for all subsequent Oracle software installations on that system. However, you can choose to create different Oracle software owner users and OSDBA groups (other than `oracle` and `dba`) for separate installations. By using different groups for different installations, members of these different groups have DBA privileges only on the associated databases, rather than on all databases on the system.

Note: If installing Oracle RAC on Microsoft Windows, Oracle Universal Installer automatically creates the `ORA_DBA` group. Also, if you install the Oracle RAC software while logged in to an account with administrative privileges, you do not need to create a separate user for the installation.

To create the required operating system user and groups on Red Hat Linux:

1. 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 by entering a command as the `root` user that is similar to the following:

   ```
   /usr/sbin/groupadd oinstall
   ```

2. Create an OSDBA group by entering a command as the `root` user that is similar to the following:

   ```
   /usr/sbin/groupadd dba
   ```

3. If the user that owns the Oracle software does not exist on your server, you must create the user. Select a user ID (UID) that is currently not in use on all the nodes in your cluster. 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 200:

   ```
   useradd -u 200 -g oinstall -G dba -d /home/oracle -r oracle
   ```

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

   ```
   passwd oracle

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

5. Repeat steps 1 through 4 on each node in your cluster as needed.

6. Verify that the attributes of the user `oracle` are identical on both `docrac1` and `docrac2`

   ```
   id oracle
   ```

   The command output should be similar to the following:

   ```
   uid=200(oracle) gid=500(oinstall) groups=500(oinstall),501(dba)
   ```

Configuring the Secure Shell

When installing Oracle RAC on UNIX and Linux platforms, the software is installed on one node, and OUI uses secure communication to copy the software binary files to the other cluster nodes. OUI uses the Secure Shell (SSH) for the communication. Various other components of Oracle RAC and Oracle Clusterware also use SSH for secure communication.

> **Note:** Oracle Net Configuration Assistant (NETCA) and Oracle Database Configuration Assistant (DBCA) require `scp` and `ssh` to be located in the path `/usr/local/bin` on the Red Hat Linux platform. If `scp` and `ssh` are not in this location, then create a symbolic link in `/usr/local/bin` to the location where `scp` and `ssh` are found.

To configure SSH, you must first create Rivest-Shamir-Adleman (RSA) keys and Digital Signature Algorithm (DSA) keys on each cluster node. After you have created the private and public keys, you copy the keys from all cluster node members into an authorized keys file that is identical on each node. When this is done, you then start the SSH agent to load the keys into memory.

> **See Also:** *Oracle Database Advanced Security Administrator's Guide* for more information about data security using encryption keys

Generating RSA and DSA Keys

Create the RSA and DSA keys on each cluster node as the first step in configuring SSH.

To configure the RSA and DSA keys on Red Hat Linux, perform the following tasks:

1. Log out and then log back in to the operating system as the `oracle` user on `docrac1`.

   Note: Do not use the `su` command to switch from the `root` user to the `oracle` user for these steps. You must completely exit your operating system session as the `root` user and start a new session as `oracle` for these steps to succeed.

2. Determine if a `.ssh` directory exists in the `oracle` user's home directory. If not, create the `.ssh` directory and set the directory permission so that only the `oracle` user has access to the directory, as shown here:

   ```
   $ ls -a $HOME
   $ mkdir ~/.ssh
   $ chmod 700 ~/.ssh
   ```

3. Create the RSA-type public and private encryption keys. Open a terminal window and run the following command:

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

   At the prompts:

   - Accept the default location for the key file by pressing the Enter key.
   - When prompted for a pass phrase, enter and confirm a pass phrase that is different from the `oracle` user's password.

   This command creates the public key in the `/home/oracle/.ssh/id_rsa.pub` file and the private key in the `/home/oracle/.ssh/id_rsa` file.

   WARNING: To protect the security of your system, never distribute the private key to anyone.

4. Create the DSA type public and private keys on both `docrac1` and `docrac2`. In the terminal window for each node, run the following command:

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

   At the prompts:

   - Accept the default location for the key file by pressing the Enter key.
   - When prompted for a pass phrase, enter and confirm a pass phrase that is different from the `oracle` user's password.

   This command creates the public key in the `/home/oracle/.ssh/id_dsa.pub` file and the private key in the `/home/oracle/.ssh/id_dsa` file.

WARNING: To protect the security of your system, never distribute the private key to anyone.

5. Repeat steps 1 through 4 on each node that you intend to add to the cluster.

Adding the Keys to an Authorized Key File

After you have generated the keys, you copy the keys for each node to an `authorized_keys` file and copy this file to all nodes in the cluster.

To add the generated keys to an authorized keys files:

1. On the local node, change directories to the `.ssh` directory in the Oracle user home directory.

   ```
   cd ~/.ssh
   ```

2. Add the RSA and DSA keys to the authorized_keys files using the following commands, then list the contents of the .ssh directory:

   ```
   $ cat id_rsa.pub >>authorized_keys
   $ cat id_dsa.pub >>authorized_keys
   $ ls
   ```

   You should see the `id_dsa.pub` and `id_rsa.pub` keys that you generated, the `id_dsa` and `id_rsa` private key files, as well as the `authorized_keys` file.

3. Use Secure Copy (SCP) or Secure FTP (SFTP) to copy the `authorized_keys` file to the `oracle` user `.ssh` directory on a remote node. The following example uses SCP to copy the `authorized_keys` file to `docrac2`, and the `oracle` user path is `/home/oracle`:

   ```
   [oracle@docrac1 .ssh]scp authorized_keys docrac2:/home/oracle/.ssh/
   The authenticity of host 'docrac2(143.46.43.101) can't be established.RSA key
   fingerprint is 7z:ez:e7:f6:f4:f2:d1:a6:f7:4e:zz:me:a7:48:ae:f6:7e.
   Are you sure you want to continue connecting (yes/no)? yes
   oracle@docrac2's password:
   ```

   You are prompted to accept an RSA or DSA key. Enter yes, and you see that the node you are copying to is added to the `known_hosts` file as shown in the preceding sample output.

   When prompted, provide the password for the `oracle` user, which should be the same on all the nodes in the cluster (Note: this is the user password, not the newly specified passphrase). The `authorized_keys` file is then copied to the remote node.

4. Using SSH, log in to the node where you copied the `authorized_keys` file, using the passphrase you created. Then change to the `.ssh` directory, and using the `cat` command, add the RSA and DSA keys for the second node to `authorized_keys` file, as demonstrated here:

   ```
   [oracle@docrac1 .ssh]$ ssh docrac2
   Enter passphrase for key '/home/oracle/.ssh/id_rsa':
   [oracle@docrac2 oracle]S cd .ssh
   [oracle@docrac2 ssh]$ cat id_rsa.pub  >> authorized_keys
   [oracle@docrac2 ssh]$ cat id_dsa.pub  >> authorized_keys
   ```

5. If you have more than two nodes in your cluster, repeat step 3 and step 4 for each node you intend to add to your cluster. Copy the most recently updated

authorized_keys file to the next node, then add the public keys for that node to the `authorized_keys` file.

6. When you have updated the `authorized_keys` file on all nodes, use SCP to copy the complete `authorized_keys` file from the last node to be updated to all the other cluster nodes, overwriting the existing version on the other nodes. For example:

```
[oracle@docrac2 .ssh]scp authorized_keys docrac1:/home/oracle/.ssh/
The authenticity of host 'docrac1(143.46.43.100) can't be established.RSA key
fingerprint is 7e:62:60:f6:f4:f2:d1:a6:f7:4e:zz:me:b9:48:dc:e3:9c.
Are you sure you want to continue connecting (yes/no)? yes
oracle@docrac2's password:
Warning: Permanently added 'docrac1,143.46.43.100' (RSA) to the list of known
hosts.
oracle@docrac1's password:
authorized_keys                          100%  1656    19.9MB.s    00:00
```

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

Configuring SSH User Equivalency

User equivalency exists in a cluster when the following occurs on *all* nodes in the cluster:

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

On Linux systems, to enable Oracle Universal Installer to use the `ssh` and `scp` commands without being prompted for a pass phrase, you must configure user SSH equivalency.

To configure user SSH equivalency on Red Hat Linux:

1. On the system where you want to run Oracle Universal Installer, log in as the `oracle` user.
2. Start the SSH agent and load the SSH keys into memory using the following commands:

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

At the prompt, enter the pass phrase for each key that you generated when configuring SSH. For example:

```
[oracle@docrac1 .ssh]$ exec /usr/bin/ssh-agent $SHELL
[oracle@docrac1 .ssh]$ /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)
```

These commands start the ssh-agent on the node, and load the RSA and DSA keys into memory so that you are not prompted to use pass phrases when issuing SSH commands

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

> **Note:** Do not close this terminal window until you have completed the installation. If you must close this terminal window before the installation is complete, repeat step 2 before starting the installation.

3. Complete the SSH configuration by using the `ssh` command to retrieve the date on each node in the cluster.

   For example, in a two-node cluster, with nodes named `docrac1` and `docrac2`, you would enter the following commands:

   ```
   $ ssh docrac1 date
   $ ssh docrac2 date
   ```

   The first time you use SSH to connect to one node from another node, you see a message similar to the following:

   ```
   The authenticity of host 'docrac1(143.46.43.100) can't be established.
   RSA key fingerprint is 7z:ez:e7:f6:f4:f2:d1:a6:f7:4e:zz:me:a7:48:ae:f6:7e.
   Are you sure you want to continue connecting (yes/no)? yes
   ```

   Enter `yes` at the prompt to continue. You should not see this message again when you connect to this node to the other node. If you see any other messages or text, apart from the date, then the installation can fail.

   If any node prompts for a password or pass phrase, then verify that the `~/.ssh/authorized_keys` file on that node contains the correct public keys. Make any changes required to ensure that only the date is displayed when you enter these commands. You should also ensure that any parts of login scripts that generate output or ask any questions are modified so that they act only when the shell is an interactive shell.

   At the end of this step, each public hostname for each member node should be registered in the `known_hosts` file for all other cluster member nodes.

Configuring the Operating System Environment

On Red Hat Linux, you run Oracle Universal Installer from the `oracle` account. Oracle Universal Installer obtains information from the environment variables configured for the `oracle` user. Prior to running OUI, you should modify the `oracle` user environment variables to configure the following:

- Set the default file mode creation mask (`umask`) to 022 in the shell startup file on Linux and UNIX systems.
- Set the `ORACLE_BASE` environment variable to the location in which you plan to install the Oracle Database software. Refer to "Choosing an Oracle Base Directory" on page 2-20 for more information about the `ORACLE_BASE` directory.

Also, if the `/tmp` directory has less than 400 MB of available disk space, but you have identified a different file system that has at least 400 MB of available space, you can set the `TEMP` and `TMPDIR` environment variables to specify the alternate temporary directory on this file system.

Prior to installing Oracle Clusterware, you can set the `ORACLE_HOME` variable to the location of the Oracle Clusterware home directory. However, you also specify the directory in which the software should be installed as part of the installation process.

After Oracle Clusterware has been installed, the `ORACLE_HOME` environment variable will be modified to reflect the value of the Oracle Database home directory.

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

Configuring the Network

Oracle Clusterware requires that you connect the nodes in the cluster to a private network by way of a private interconnect. Each node in the cluster must also be accessible by way of the public network.

To configure the network and ensure that each node in the cluster is able to communicate with the other nodes in the cluster:

1. Determine your cluster name. The cluster name should satisfy the following conditions:
   - The cluster name is globally unique throughout your host domain.
   - The cluster name is at least 1 character long and less than 15 characters long.
   - The cluster name consists of the same character set used for host names: underscores (_), hyphens (-), and single-byte alphanumeric characters (a to z, A to Z, and 0 to 9).
   - If you use third-party vendor clusterware, then Oracle recommends that you use the vendor cluster name.
2. Determine the public node names, private node names, and virtual node names for each node in the cluster.
   - For the public node name, use the primary host name of each node. In other words, use the name displayed by the `hostname` command. This node name can be either the permanent or the virtual host name, for example: `docrac1`.
   - Determine a private node name or private IP address for each node. The private IP address is an address that is accessible only by the other nodes in this cluster. Oracle Database uses private IP addresses for internode, or instance-to-instance Cache Fusion traffic. Oracle recommends that you provide a name in the format *`public_hostname`*`-priv`, for example: `docrac1-priv`.
   - Determine a virtual host name for each node. A virtual host name is a public node name that is used to reroute client requests sent to the node if the node is down. Oracle Database uses virtual IP addresses for client-to-database connections, so the VIP address must be publicly accessible. Oracle recommends that you provide a name in the format *`public_hostname`*`-vip`, for example: `docrac1-vip`.
3. Identify the interface names and associated IP addresses for all network adapters by running 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 that you want to specify as a public or private network interface.

Note: When you install Oracle Clusterware and Oracle RAC, you will require this information.

---

4. On each node in the cluster, assign a public IP address with an associated network name to one network adapter, and a private IP address with an associated network name to the other network adapter.

   The public name for each node should be registered with your domain name system (DNS). If you do not have an available DNS, then record the network name and IP address in the system hosts file, `/etc/hosts`. Use the `/etc/hosts` file on each node to associate the private network name for that host with its private IP address.

   You can test whether or not an interconnect interface is reachable using a `ping` command.

5. On each node in the cluster, configure a third IP address that will serve 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. If you do not have an available DNS, then record the virtual host name and IP address in the system hosts file, `/etc/hosts`.

6. When you complete the network configuration, the IP address and network interface configuration should be similar to what is shown in the following table (your node names and IP addresses might be different):

Node	Node Name	Type	IP Address	Registered in
docrac1	docrac1	Public	143.46.43.100	DNS (if available, else the hosts file)
docrac1	docrac1-vip	Virtual	143.46.43.104	DNS (if available, else the hosts file)
docrac1	docrac1-priv	Private	10.10.10.11	Hosts file
docrac2	docrac2	Public	143.46.43.101	DNS (if available, else the hosts file)
docrac2	docrac2-vip	Virtual	143.46.43.105	DNS (if available, else the hosts file)
docrac2	docrac2-priv	Private	10.10.10.12	Hosts file

After you have completed the installation process, you will configure clients to use either the virtual IP address or the network name associated with the virtual IP address.

Verifying the Network Configuration

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

To verify the network configuration on a two-node cluster that is running Red Hat 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 on both `docrac1` and `docrac2`.

   In this example, `eth0` is used for the public network and `eth1` is used for the private network, which is used for Cache Fusion communications.

```
# /sbin/ifconfig

eth0      Link encap:Ethernet  HWaddr 00:0E:0C:08:67:A9
          inet addr: 143.46.43.100   Bcast:143.46.43.255   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

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:52012956 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52012956 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:905082901 (863.1 Mb)  TX bytes:905082901 (863.1 Mb)
```

2. As the `root` user, verify that the `/etc/hosts` file on the node `docrac1` contains the host IP addresses, virtual IP addresses, and private network IP addresses from both nodes in the cluster, as follows:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain       localhost
143.46.43.100   docrac1.mycompany.com          docrac1
143.46.43.104   docrac1-vip.mycompany.com      docrac1-vip
10.10.10.11     docrac1-priv

143.46.43.101   docrac2.mycompany.com          docrac2
143.46.43.105   docrac2-vip.mycompany.com      docrac2-vip
10.10.10.12     docrac2-priv
```

   If the `/etc/hosts` file is missing any of the preceding information, then edit the file to add the necessary information.

   After the `/etc/hosts` file is configured on `docrac1`, edit the `/etc/hosts` file on `docrac2` so it contains the same information for the cluster IP addresses.

3. As the `root` user, verify the network configuration by using the `ping` command to test the connection from `docrac1` from `docrac2` and the reverse. As the `root` user, run the following commands on each node:

```
# ping -c 3 docrac1.mycompany.com
# ping -c 3 docrac1
# ping -c 3 docrac1-priv

# ping -c 3 docrac2.mycompany.com
# ping -c 3 docrac2
# ping -c 3 docrac2-priv
```

You will not be able to discover the nodes using the `ping` command for the virtual IPs (`docrac1-vip`, `docrac2-vip`) until after Oracle Clusterware is installed and running. If the `ping` commands for the public or private addresses fail, resolve the issue before you proceed.

4. 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 Red Hat Linux Help utility.

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:

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

You must ensure that you have a certified combination of the operating system and the Oracle Database software by referring to Oracle Meta*Link* certification, which is located at the following Web site:

`http://metalink.oracle.com`

You can find this by clicking **Certify & Availability** and then selecting **1.View Certifications by Product**.

Note: Oracle Universal Installer verifies that your system meets the listed requirements. Check the requirements before you start Oracle Universal Installer, to ensure your system will meet the requirements.

Setting the Time on Both Nodes

Before starting the installation, ensure that the date and time settings on both nodes are set as closely as possible to the same date and time. Oracle strongly recommends using the Network Time Protocol (NTP) feature of most operating systems for this purpose.

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.

Configuring Kernel Parameters

Oracle Universal Installer 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 in this section, then *do not* modify the parameter setting.

See Also: *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform for more information about tuning kernel parameters

Performing Platform-Specific Configuration Tasks

You may be required to perform special configurations 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 Red Hat Enterprise Linux AS 2.1 (Itanium), SUSE Linux Enterprise Server 9, or Red Hat Enterprise Linux 4.
- Configure the hangcheck-timer module on Red Hat Linux 3.0, SUSE 8, Red Hat Linux 4.0 and SUSE 9 systems.
- Set shell limits for the `oracle` user on Red Hat Linux systems to increase the number of files and processes available to Oracle Clusterware and Oracle RAC.
- Start the Telnet service on Microsoft Windows.
- Create X library symbolic links on HP-UX.
- Configure network tuning parameters on AIX.

See Also: *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform for more information about tasks that are specific to your platform

Configuring Installation Directories and Shared Storage

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

- Deciding on a Shared Storage Solution
- Configuring the Raw Storage Devices and Partitions
- Choosing an Oracle Base Directory
- Choosing an Oracle Clusterware Home Directory

Deciding on a Shared Storage Solution

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:

- A supported cluster file system, such as Oracle Cluster File System (OCFS) for Microsoft Windows and Linux or General Parallel File System (GPFS) on IBM platforms
- Network file system (NFS), which is not supported on AIX, POWER, or on IBM zSeries-based Linux
- ASM for Oracle Database files (strongly recommended)

> **Note:** Oracle Clusterware files cannot be stored in ASM.

For all installations, you must choose the storage option that you want to use for Oracle Clusterware files and Oracle Database files.

If you do not have an NFS or cluster file system available, you can use raw devices to store the Oracle Clusterware files. A **raw device** is a disk drive that does not yet have a file system set up. Raw devices have device names in the form `/dev/raw/raw`*n*, where *n* is a number that identifies the raw device. Raw devices are commonly used for Oracle RAC because they enable the sharing of disks.

> **Note:** For the most up-to-date information about supported storage options for RAC installations, refer to the Certify pages on OracleMeta*Link*
>
> http://metalink.oracle.com

If you decide to use OCFS to store the Oracle Clusterware files, you must use the proper version of OCFS for your operating system version. OCFS v1 works with RedHat Linux 2.4 and OCFS v2 works with RedHat Linux 2.6 The examples in this guide, which are based on Red Hat Linux, use raw partitions to store the Oracle Clusterware files and Oracle ASM to store the Oracle database files. The Oracle Clusterware and Oracle Database software will be installed on disks local to each node, not on a shared file system.

The following section describes how to create the raw partitions for the Oracle Clusterware files on Red Hat Linux.

> **See Also:** *Oracle Clusterware and Oracle Real Application Clusters Installation and Configuration Guide* for your platform if you are using a cluster file system or NFS

Configuring the Raw Storage Devices and Partitions

Physical disk space needs to be allocated in partitions on the disks where you want to set up raw devices. You use an operating system command to create the raw partitions. You can create multiple partitions on a single disk.

Before you install Oracle Clusterware, you will need to configure 5 raw partitions, each 256 MB in size, for storing the Oracle Cluster Registry (OCR), a duplicate OCR file on a different disk, referred to as the OCR mirror, and three voting disks. If you plan to use raw devices for storing the database files, you will need to create additional raw partitions for each tablespace, online redo log file, control file, server parameter file (SPFILE) and password file.

To configure raw partitions for Oracle Clusterware files on Red Hat Linux:

1. To identify the device name for the disks that you want to use, enter the following command on the first node in your cluster, for example, `docrac1`:

   ```
   # /sbin/fdisk -l
   ```

   You can create the required raw partitions either on new devices that you added or on previously partitioned devices that have unpartitioned available space. To identify devices that have unpartitioned available space, examine the start and end cylinder numbers of the existing partitions and determine whether or not the device contains unused cylinders.

2. As the `root` user, configure storage for the OCR, the voting disk files, and the database files. If you are using Internet small computer system interface (iSCSI) storage, provide a mapping from a block device to a character device by adding entries in the `/etc/sysconfig/rawdevices` file.

   Create two raw partitions 256 MB in size for the OCR and its mirror, and three partitions 256 MB in size for the Oracle Clusterware voting disks.

   To create raw partitions on a device, as the `root` user, enter a command similar to the following, where *`devicename`* is the name of a raw device:

   ```
   # /sbin/fdisk devicename
   ```

   Use the following guidelines when creating partitions:

   - Use the `p` command to list the partition table of the device.
   - Use the `n` command to create a partition.
   - After you have created the required partitions on this device, use the `w` command to write the modified partition table to the device.
   - Refer to the `fdisk` entry in the Linux Help system for more information about creating partitions.

   The following example uses `fdisk` to create a 256 MB partition on the raw device, `/dev/sdb`, on the first node. This partition, or slice, will be used for the OCR disk. You will create another 256 MB partition on a different disk and disk controller for the OCR mirror. Each file should be on a different disk and disk controller.

   ```
   # /sbin/fdisk /dev/sdb
   Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
   disklabel
   Building a new DOS disklabel. Changes will remain in memory only, until you
   decide to
   write them. After that, of course, the previous content won't be recoverable.

   Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

   Command (m for help): p

   Disk /dev/sdb: 1073 MB, 107341824 bytes
   34 heads, 61 sectors/track, 1011 cylinders
   Units = cylinders of 2074 * 512 = 1061888 bytes

      Device boot           Start       End      Blocks     ID  System

   Command (m for help): n
   Command action
     e  extended
   ```

```
  p  primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1011, default 1):
Using default value 1
Last cylinder of +size or +sizeM or +sizeK (1-1011, default 1011): +256M

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
#
```

3. Enter the following command to create a 256 MB partition on the second raw device, `/dev/sdc`. This partition will be used for the OCR mirror. Use the same prompts as shown in step 2. Put each voting disk file on a different disk and controller.

```
fdisk /dev/sdc
```

4. Use the `fdisk` command to create 256 MB partitions on the raw devices `/dev/sdd`, `/dev/sde` and `/dev/sdf`. These partitions will be used for the voting disk files. Put each file on a different disk and controller.

```
fdisk /dev/sdd

fdisk /dev/sde

fdisk /dev/sdf
```

   Each time you run the command, use the same responses as in step 2.

5. As the `root` user on `docrac1`, edit the `/etc/sysconfig/rawdevices` file and add the mappings for the raw devices used by Oracle Clusterware. The following example also shows the mappings for ASM:

```
# raw device bindings
# format:  <rawdev> <major> <minor>
#          <rawdev> <blockdev>
# example: /dev/raw/raw1 /dev/sda1
#          /dev/raw/raw2 8 5
#OCR Devices
/dev/raw/raw1   /dev/sdb1
/dev/raw/raw2   /dev/sdc1
#Voting Disk Devices
/dev/raw/raw3   /dev/sdd1
/dev/raw/raw4   /dev/sde1
/dev/raw/raw5   /dev/sdf1
#ASM Disk Devices
/dev/raw/raw6   /dev/sdg
/dev/raw/raw7   /dev/sdh
/dev/raw/raw8   /dev/sdi
```

   You have to create at least two partitions, one for the OCR, and the other for the voting disk. In steps 2 through 4, you created two OCR files and three voting disk files to improve the availability of the Oracle RAC database. The minimum size for a voting disk file is 25 MB.

6. As the `root` user, on the node `docrac1`, enable the raw devices so that the mappings become effective at the operating system level using the following command:

   ```
   service rawdevices start
   ```

7. On the node `docrac2`, as the `root` user, for each of the disks you used in the previous steps 2, 3, and 4 run the `partprobe` command. For example, if you configured disks `/dev/sdb`, `/dev/sdc`, `/dev/sdd`, `/dev/sde` and `/dev/sdf` in the previous commands, then you would run the following commands:

   ```
   /sbin/partprobe /dev/sdb
   /sbin/partprobe /dev/sdc
   /sbin/partprobe /dev/sdd
   /sbin/partprobe /dev/sde
   /sbin/partprobe /dev/sdf
   ```

   This forces the operating system on the other node in the cluster to refresh its picture of the shared disk partitions.

8. Repeat step 5 as the `root` user on `docrac2`.

9. As the `root` user, on the node `docrac2`, start the raw devices so they are visible at the operating system level using the following command:

   ```
   /sbin/service rawdevices start
   ```

10. As the `root` user, on each node in the cluster, enter commands similar to the following to set the owner, group, and permissions on the newly created device files:

    ```
    chown  root:oinstall  /dev/raw/raw[1-2]# for raw1 through raw2
    chown  oracle:oinstall /dev/raw/raw[3-5] # for raw3 through raw5
    chmod  640  /dev/raw/raw[1-2]
    chmod  640  /dev/raw/raw[3-5]
    chown oracle:dba /dev/sdg
    chown oracle:dba /dev/sdh
    chown oracle:dba /dev/sdi
    chmod 660 /dev/sdg
    chmod 660 /dev/sdh
    chmod 660 /dev/sdi
    ```

    Repeat this step on the node `docrac2`.

Configuring Raw Devices on Red Hat Enterprise Linux 4.0

Starting with the 2. 6 Linux kernel distributions, raw devices are not supported by default in the kernel. However, Red Hat Enterprise Linux 4.0 continues to provide raw device support.

To configure raw devices if you are using Red Hat Enterprise Linux 4.0:

1. To confirm that raw devices are enabled, enter the following command:

   ```
   # chkconfig --list
   ```

2. Scan the output for raw devices. If you do not find raw devices, then use the following command to enable the raw device service:

   ```
   # chkconfig --level 345 rawdevices on
   ```

3. After you confirm that the raw devices service is running, you should change the default ownership of raw devices. When you restart a Red Hat Enterprise Linux 4.0 system, ownership and permissions on raw devices revert by default to the `root` user. If you are using raw devices with this operating system for your Oracle Clusterware files, then you need to override this default.

   To ensure correct ownership of these devices when the operating system is restarted, create a new file in the `/etc/udev/permissions.d` directory, called `oracle.permissions`, and enter the raw device permissions information. Using the example device names discussed in step 5 of the previous section, the following is an example of the contents of `/etc/udev/permissions.d/oracle.permissions`:

   ```
   # OCR
   raw/raw[12]:root:oinstall:0640
   # Voting Disks
   raw/raw[3-5]:oracle:oinstall:0640
   # ASM
   raw/raw[67]:oracle:dba:0660
   ```

4. After creating the `oracle.permissions` file, the permissions on the raw devices are set automatically the next time the system is restarted. To set permissions to take effect immediately, without restarting the system, use the `chown` and `chmod` commands:

   ```
   chown root:oinstall /dev/raw/raw[12]
   chmod 640 /dev/raw/raw[12]
   chown oracle:oinstall /dev/raw/raw[3-5]
   chmod 640 /dev/raw/raw[3-5]
   chown oracle:dba /dev/raw/raw[67]
   chmod 660 /dev/raw/raw[67]
   ```

Choosing an Oracle Base Directory

OUI creates the Oracle base directory for you in the location you specify. The Oracle base directory (`ORACLE_BASE`) acts as a top-level directory for Oracle software installations. Optimal Flexible Architecture (OFA) guidelines recommend that you use a path similar to the following for the Oracle base directory:

```
/mount_point/app/oracle
```

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.

The file system that you use for the Oracle base directory must have at least 1.5 GB of available disk space for installing the Oracle Database software. The path to the Oracle base directory must be the same on all nodes.

For Red Hat Linux systems, you can use the `df -h` command to determine the available disk space on each mounted file system. Choose a file system that has sufficient available space. For the sample installation described in this guide, the chosen mount point must have at least 3 GB of available space, for installing Oracle RAC and Oracle ASM in separate home directories. The examples in this guide use `/opt/oracle/10gR2` for the Oracle base directory.

Choosing an Oracle Clusterware Home Directory

Oracle Universal Installer (OUI) installs Oracle Clusterware into a directory structure referred to as *`CRS_home`*. This home is separate from the home directories for other

Oracle products installed on the same server. OUI creates the Oracle Clusterware home directory for you. Before you start the installation make sure that you have sufficient disk space on a file system for the Oracle Clusterware directory, and that the Oracle Clusterware home directory is owned by `root`.

The file system that you use for the Oracle Clusterware home directory must have at least 120 MB of available disk space. The path to the Oracle Clusterware home directory must be the same on all nodes.

For Red Hat Linux, you can use the `df -h` command to determine the available disk space on each mounted file system. Choose a file system that has appropriate available space. For the examples in this guide, the directory `/opt/oracle/crs` will be used for the Oracle Clusterware home directory.

Note: Ensure the Oracle Clusterware home directory is not a subdirectory of the `ORACLE_BASE` directory.

3

Installing and Configuring Oracle Clusterware and Oracle RAC

This chapter explains how to install Oracle Real Application Clusters (Oracle RAC) using Oracle Universal Installer (OUI). You must install Oracle Clusterware before installing Oracle RAC. After your Oracle Clusterware is operational, you can use OUI to install the Oracle Database software with the Oracle RAC components.

The example Oracle RAC environment described in this guide uses Oracle Automatic Storage Management (ASM), so this chapter also includes instructions on how to install ASM in its own home directory.

This chapter includes the following sections:

- Preparing the Oracle Media Installation File
- Installing Oracle Clusterware 10g
- Configuring Automatic Storage Management in an ASM Home Directory
- Installing the Oracle Database Software and Creating a Cluster Database
- Performing Postinstallation Tasks
- Converting an Oracle Database to an Oracle RAC Database

Preparing the Oracle Media Installation File

Oracle Clusterware is not installed as part of Oracle Database 10*g*, but is installed from the Oracle Clusterware installation media. Because Oracle Clusterware works closely with the operating system, system administrator access is required for some of the installation tasks. In addition, some of the Oracle Clusterware processes must run as the special operating system user, `root`.

The Oracle RAC Database software is installed from the Oracle Database 10*g* installation media. By default, the standard Oracle Database 10*g* 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 that is referred to as *`Oracle_home`*. This home is separate from the home directories of other Oracle software products installed on the same server.

If the Oracle Clusterware installation software and Oracle Database installation software are in ZIP files, create a staging directory on one node, for example, `docrac1`, to store the unzipped files, as shown here:

```
mkdir -p /stage/oracle/10.2.0
```

Copy the ZIP files to this staging directory. For example, if the files were downloaded to a directory named `/home/user1`, and the ZIP files are named `10201_clusterware_linux32.zip` and `10201_database_linux32.zip`, you would you use the following commands to move the ZIP files to the staging directory:

```
cd /home/user1
cp 10201_clusterware_linux32.zip /stage/oracle/10.2.0
cp 10201_database_linux32.zip /stage/oracle/10.2.0
```

Then, as the `oracle` user on `docrac1`, unzip the Oracle media, as shown in the following example:

```
cd /stage/oracle/10.2.0
unzip 10201_clusterware_linux32.zip
unzip 10201_database_linux32.zip
```

If you have the Oracle Clusterware and Oracle Database software on CDs, insert the distribution media for the database into a disk drive on your computer. Make sure the disk drive has been mounted at the operating system level.

Installing Oracle Clusterware 10*g*

The following topics describe the process of installing Oracle Clusterware:

- Configuring the Operating System Environment
- Verifying the Configuration Using the Cluster Verification Utility
- Using Oracle Universal Installer to Install Oracle Clusterware
- Completing the Oracle Clusterware Configuration

Configuring the Operating System Environment

You run Oracle Universal Installer from the `oracle` user account. However, before you start Oracle Universal Installer you must configure the environment of the `oracle` user. You must set the `ORACLE_SID` and `ORACLE_BASE` environment variables to the desired values for your environment.

For example, if you want to create an Oracle database named `sales` on the mount point directory `/opt/oracle`, you would set `ORACLE_SID` to `sales` and `ORACLE_BASE` to the directory `/opt/oracle/10gR2`.

To modify the user environment on Red Hat Linux:

1. As the `oracle` user, modify the user profile in the `/home/oracle` directory on both nodes using the following commands:

   ```
   [oracle] $ cd $HOME
   [oracle] $ vi .bash_profile
   ```

   Add the following lines at the end of the file:

   ```
   export ORACLE_SID=sales
   export ORACLE_BASE=/opt/oracle/10gR2
   export ORACLE_HOME=/opt/oracle/crs
   export PATH=$PATH:$ORACLE_HOME/bin
   ```

   In the previous example, the `ORACLE_HOME` variable has been set to the location of the Oracle Clusterware home directory. After Oracle Clusterware has been

installed the `ORACLE_HOME` environment variable will be modified to reflect the value of the Oracle Database home directory.

2. Read and execute the changes made to the `.bash_profile` file:

```
source .bash_profile
```

Verifying the Configuration Using the Cluster Verification Utility

If you have not configured your nodes, network, and operating system correctly, your installation of the Oracle Clusterware or Oracle Database 10g software will not complete successfully.

As the `oracle` user, change directories to the staging directory for the Oracle Clusterware software, or to the mounted installation disk. Then, enter the following command to verify your hardware and operating system setup, where *`staging_area`* is the location of the installation media (for example, `/home/oracle/downloads/10gR2/10.2.0` or `/dev/dvdrom`):

```
[oracle] $ cd /staging_area/clusterware/cluvfy
[oracle] $ ./runcluvfy.sh stage -pre crsinst -n docrac1,docrac2 -verbose
```

The preceding command instructs the CVU to verify that the system meets all the criteria for an Oracle Clusterware installation. It checks that all the nodes are reachable from the local nodes, proper user equivalence exists, connectivity exists between all the nodes through the public and private interconnects, the user has proper permissions to install the software, and that all system requirements (including kernel version, kernel parameters, memory, swap space, temporary directory space, required software packages) are met.

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about resolving the CVU errors

Using Oracle Universal Installer to Install Oracle Clusterware

As the `oracle` user on the `docrac1` node, install Oracle Clusterware. Note that OUI uses Secure Shell (SSH) to copy the binary files from `docrac1` to `docrac2` during the installation.

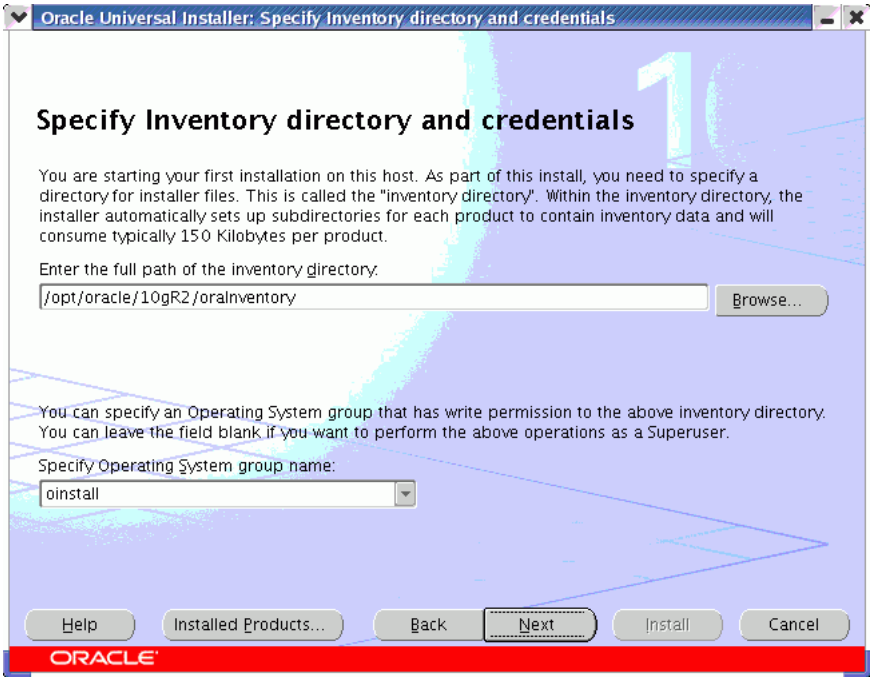
> **Note:** If you are installing Oracle Clusterware on a server that already has a single-instance Oracle Database 10g installation, then stop the existing ASM instances, if any. After Oracle Clusterware is installed, start up the ASM instances again. When you restart the single-instance Oracle database and then the ASM instances, the ASM instances use the Cluster Synchronization Services Daemon (CSSD) instead of the daemon for the single-instance Oracle database.

To install Oracle Clusterware:

1. Use the following command to start Oracle Universal Installer, where *`staging_area`* is the location of the staging area on disk, or the location of the mounted installation disk:

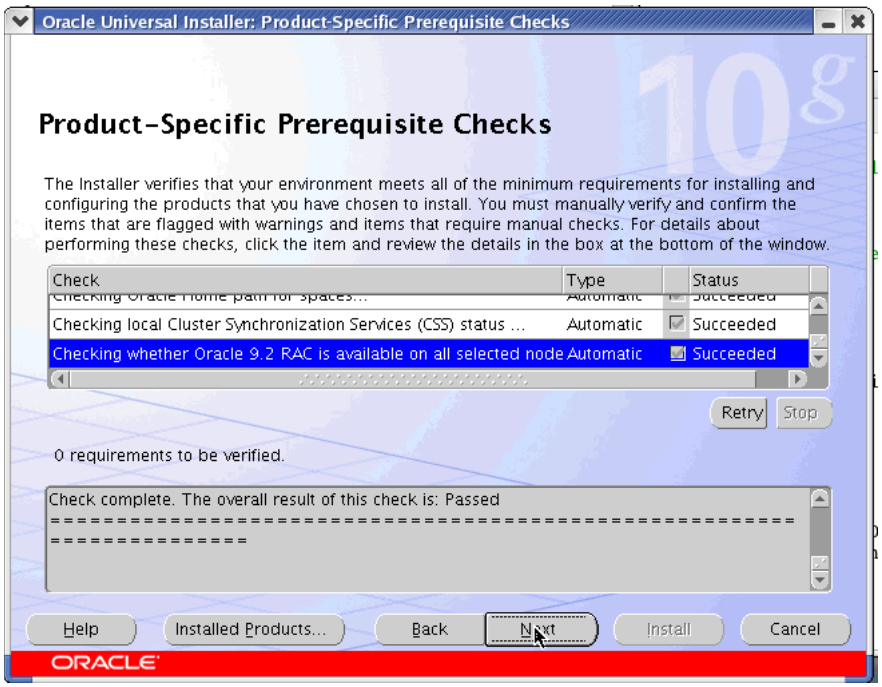
```
cd /staging_area/clusterware
./runInstaller
```

2. The OUI Welcome window appears. Click **Next**.
3. If you have not installed any Oracle software previously on this server, the Specify Inventory directory and credentials window appears. The path displayed for the inventory directory should be the `oraInventory` subdirectory of your Oracle base directory. For example, if you set the `ORACLE_BASE` environment variable to `/opt/oracle/10gR2` before starting OUI, then the path displayed is `/opt/oracle/10gR2/oraInventory`. For the operating system group name, choose `oinstall`. Click **Next**.



4. The Specify Home Details window appears. Accept the default value for the Name field, which is the name of the Oracle home directory for this product. For the Path field, click **Browse** to go to and select the directory `/opt/oracle/crs`, if this path is not already displayed.

   After you have selected the path, click **Next**.
5. The next window, Product-Specific Prerequisite Checks, appears after a short period of time. When you see the message "Check complete. The overall result of this check is: Passed", as shown in the following screen shot, click **Next**.



6. The Specify Cluster Configuration window appears.

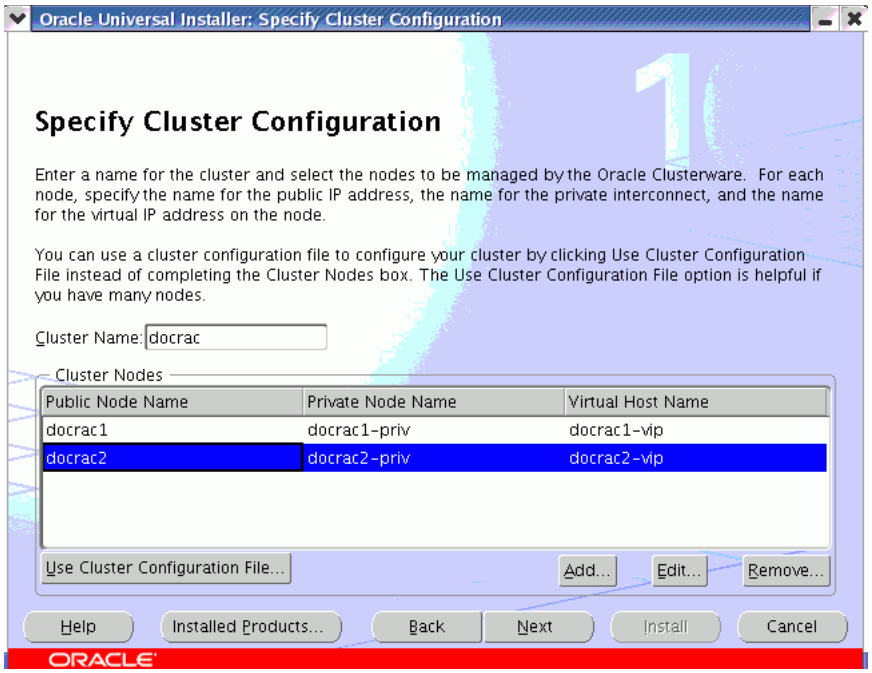
   Change the default cluster name from `crs` to a name that is unique throughout your entire enterprise network. For example, you might choose a name that is based on the node names' common prefix. This guide will use the cluster name `docrac`.

   The local node, `docrac1`, appears in the Cluster Nodes section. If the cluster node names include the domain name, click **Edit** and remove the domain name from the public, private, and virtual node names. For example, if the node name is `docrac1`, edit the entries so that they are displayed as `docrac1`, `docrac1-priv`, and `docrac1-vip`. When you have finished removing the domain names in the "Modify a node in the existing cluster" window, click **OK**.

   When you are returned to the Specify Cluster Configuration window, click **Add**.

   In the "Add a new node to the existing cluster" dialog window, enter the second node's public name (`docrac2`), private name (`docrac2-priv`), and virtual IP name (`docrac2-vip`), then click **OK**.

   The Specify Cluster Configuration window now displays both nodes in the Cluster Nodes section.

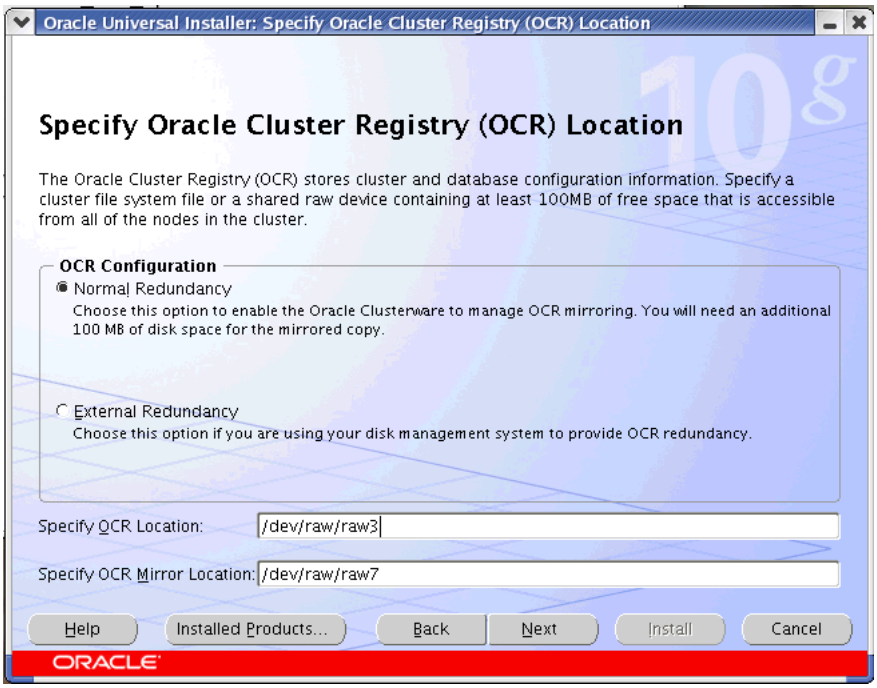


Click **Next**.

7. The Specify Network Interface Usage window appears. Verify `eth0` and `eth1` are configured correctly (proper subnet and interface type displayed), then click **Next**.

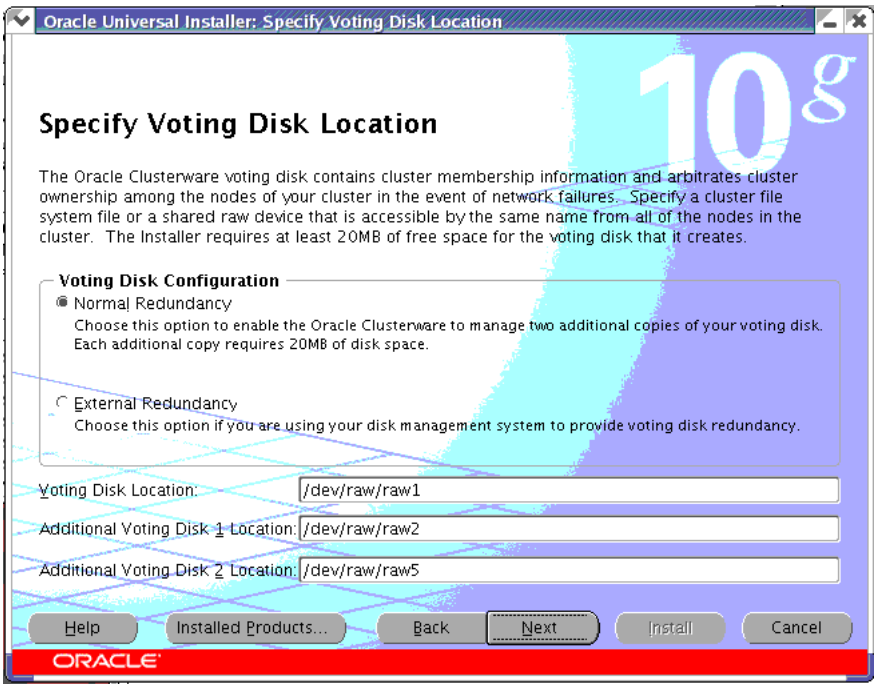
   The Specify Oracle Cluster Registry (OCR) Location window appears.

8. Choose **Normal Redundancy** for the OCR Configuration. You will be prompted for two file locations. In the Specify OCR Location field enter the name of the device configured for the first OCR file. For example, `/dev/raw/raw1`. In the Specify OCR Mirror Location field, enter the name of the device configured for the OCR mirror file, for example `/dev/raw/raw2`. When finished, click **Next**. During installation, the OCR data will be written to the specified locations.



The Specify Voting Disk Location window appears.

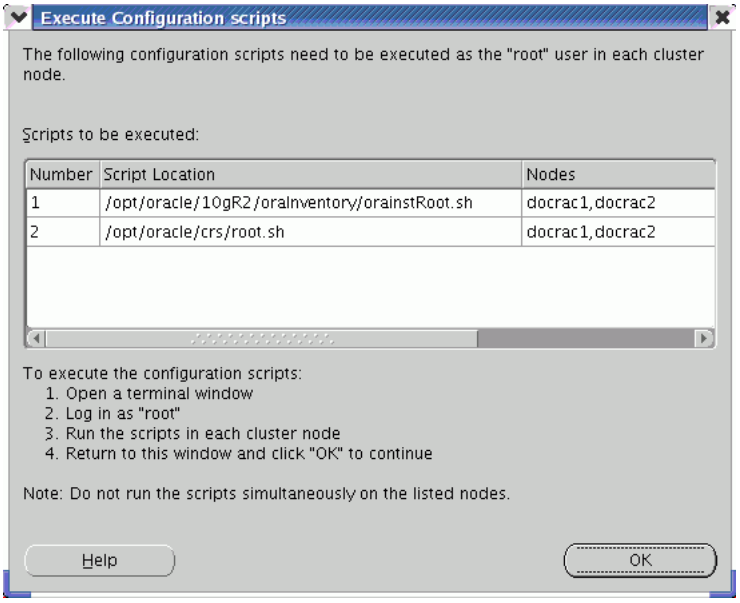
9. Select **Normal Redundancy** for the voting disk location. You will be prompted for three file locations. For the Voting Disk Location, enter the name of the device configured for the first voting disk file, for example, `/dev/raw/raw3`. Repeat this process for the other two Voting Disk Location fields. When finished, click **Next**.



10. The OUI Summary window appears. Review the contents of the Summary window and then click **Install**.

OUI displays a progress indicator during the installation process.

11. During the installation process, the Execute Configuration Scripts window appears. Do *not* click **OK** until you have run the scripts.



The Execute Configuration Scripts window shows configuration scripts, and the path where the configuration scripts are located. Run the scripts on all nodes as directed, in the order shown. For example, on Red Hat 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 `docrac1`, open a terminal window, and enter the following commands:

```
[oracle@docrac1 oracle]$ cd /opt/oracle/10gR2/oraInventory
[oracle@docrac1 oraInventory]$ su
```

b. Enter the password for the `root` user, and then enter the following command to run the first script on `docrac1`:

```
[root@docrac1 oraInventory]# ./orainstRoot.sh
```

c. After the `orainstRoot.sh` script finishes on `docrac1`, open another terminal window, and as the `oracle` user, enter the following commands:

```
[oracle@docrac1 oracle]$ ssh docrac2
[oracle@docrac2 oracle]$ cd /opt/oracle/10gR2/oraInventory
[oracle@docrac2 oraInventory]$ su
```

d. Enter the password for the `root` user, and then enter the following command to run the first script on `docrac2`:

```
[root@docrac2 oraInventory]# ./orainstRoot.sh
```

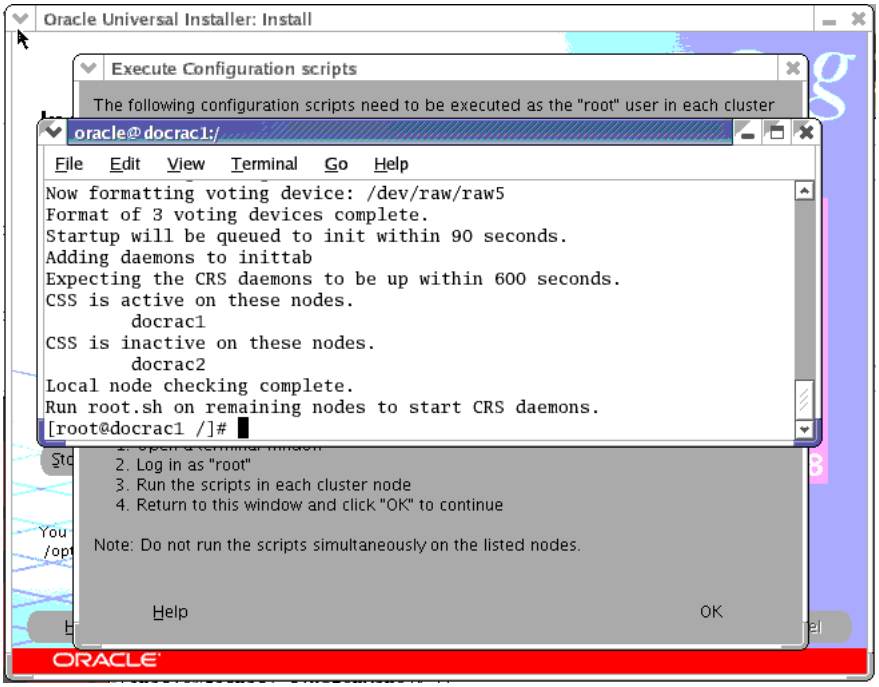
e. After the `orainstRoot.sh` script finishes on `docrac2`, go to the terminal window you opened in step b. As the `root` user on `docrac1`, enter the following commands to run the second script, `root.sh`:

```
[root@docrac1 oraInventory]# cd /opt/oracle/crs
[root@docrac1 crs]# ./root.sh
```

Note: Do not attempt to run the `root.sh` script on other nodes. Wait until the script finishes running on the local node.

At the completion of this script, the following message is displayed:

```
Local node checking complete.
Run root.sh on remaining nodes to start CRS daemons.
```



f. After the `root.sh` script finishes on `docrac1`, go to the terminal window you opened in step c. As the `root` user on `docrac2`, enter the following commands:

```
[root@docrac2 oraInventory]# cd /opt/oracle/crs
[root@docrac2 crs]# ./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**.

12. The Configuration Assistants window appears. When the configuration assistants finish, OUI displays the End of Installation window. Click **Exit** to complete the installation process.

If you encounter any problems, refer to the configuration log for information. The path to the configuration log is displayed on the Configuration Assistants window.

Completing the Oracle Clusterware Configuration

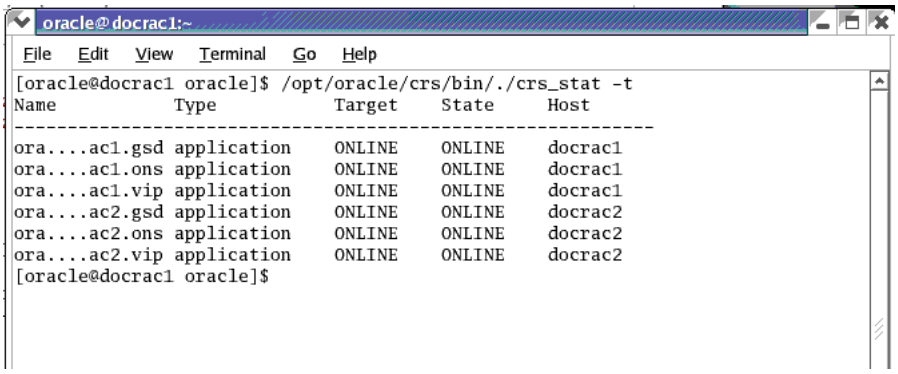
After you have installed Oracle Clusterware, verify that the node applications are running. Depending on which operating system you use, you may need to perform some postinstallation tasks to configure the Oracle Clusterware components properly.

To complete the Oracle Clusterware configuration on Red Hat Linux:

1. As the `oracle` user on `docrac1`, check the status of the clusterware targets by entering the following command:

   ```
   /opt/oracle/crs/bin/./crs_stat -t
   ```

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



2. If you are using Red Hat Linux 3.0, then, for each raw device used to store files for Oracle Clusterware, you must add two entries in the `/etc/rc.d/rc.local` file.

   The following table shows examples of the entries you must add for each file type, where `oracle` is the Oracle software owner, `oinstall` is the Oracle install group, `dba` is the privileged Oracle user group, `/dev/raw/raw#` is an individual device file, and `/dev/`*`name`* is a raw device name:

File Type	Entries to Add
OCR	chown root:oinstall /dev/raw/raw#
	chmod 640 /dev/raw/raw#
Voting disk	chown oracle:oinstall /dev/raw/raw#
	chmod 640 /dev/raw/raw#
ASM disk	chown oracle:dba /dev/*name*
	chmod 660 /dev/*name*

   Using the example raw partitions and devices listed in this guide, you would log in as `root` and insert the following at the end of the `/etc/rc.d/rc.local` file on both nodes `docrac1` and `docrac2`, so that the permissions are set correctly when the nodes are restarted:

   ```
   chown root:oinstall /dev/raw/raw1
   chown root:oinstall /dev/raw/raw2
   chown oracle:oinstall /dev/raw/raw3
   chown oracle:oinstall /dev/raw/raw4
   chown oracle:oinstall /dev/raw/raw5
   ```

```
chmod 640 /dev/raw/raw1
chmod 640 /dev/raw/raw2
chmod 640 /dev/raw/raw3
chmod 640 /dev/raw/raw4
chmod 640 /dev/raw/raw5

chown oracle:dba /dev/sdg
chown oracle:dba /dev/sdh
chown oracle:dba /dev/sdi
chmod 660 /dev/sdg
chmod 660 /dev/sdh
chmod 660 /dev/sdi
```

If you are using Red Hat Enterprise Linux 4.0, then ownership of the raw devices after restart was configured in the previous chapter using the `udev` utility in the section titled "Configuring the Raw Storage Devices and Partitions" on page 2-16.

Configuring Automatic Storage Management in an ASM Home Directory

This section explains how to install the Oracle ASM software in its own home directory. Installing ASM in its own home directory enables you to keep the ASM home separate from the database home directory (`ORACLE_HOME`). By using separate home directories, you can upgrade and patch ASM and the Oracle Database software independently, and you can deinstall Oracle Database software without affecting the ASM instance.

As the `oracle` user, install ASM by installing the Oracle Database 10*g* Release 2 software on the `docrac1` node. Note that the Installer copies the binary files from `docrac1` to `docrac2` during the installation.

To install Oracle ASM in a home directory separate from the home directory used by Oracle Database:

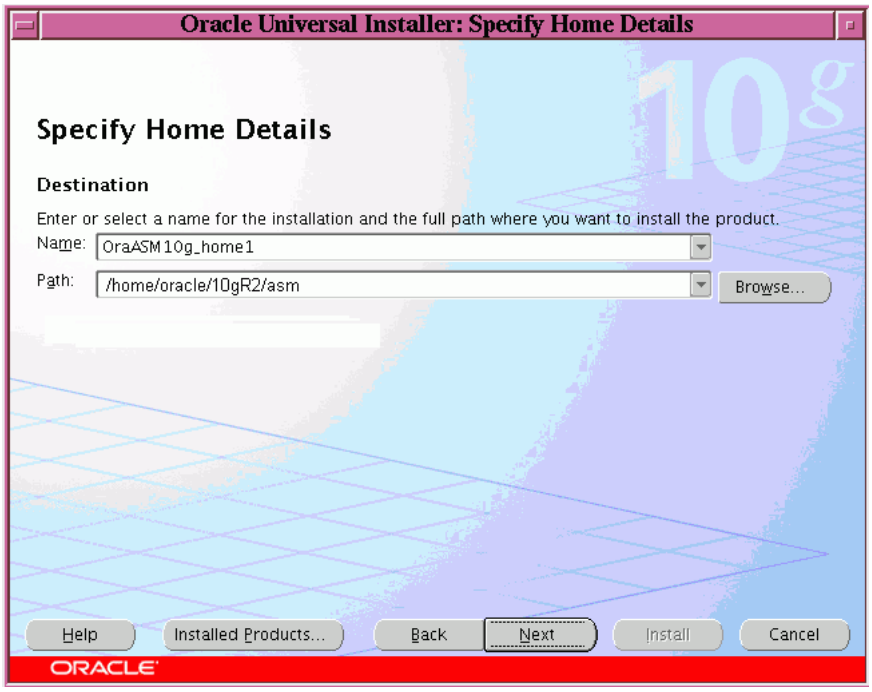
1. Use the following commands to start OUI, where *`staging_area`* is the location of the staging area on disk, or the location of the mounted installation disk:

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

   When you start Oracle Universal Installer, the Welcome window appears. Click **Next**.

2. The Select Installation Type window appears. Select either Enterprise Edition or Standard Edition and then click **Next**.

3. In the Specify Home Details window, specify a name for the ASM Home directory, for example, `OraASM10g_home`. Select a directory that is a subdirectory of your Oracle Base directory, for example, `/opt/oracle/10gR2/asm`. Click **Browse** to change the directory in which ASM will be installed.



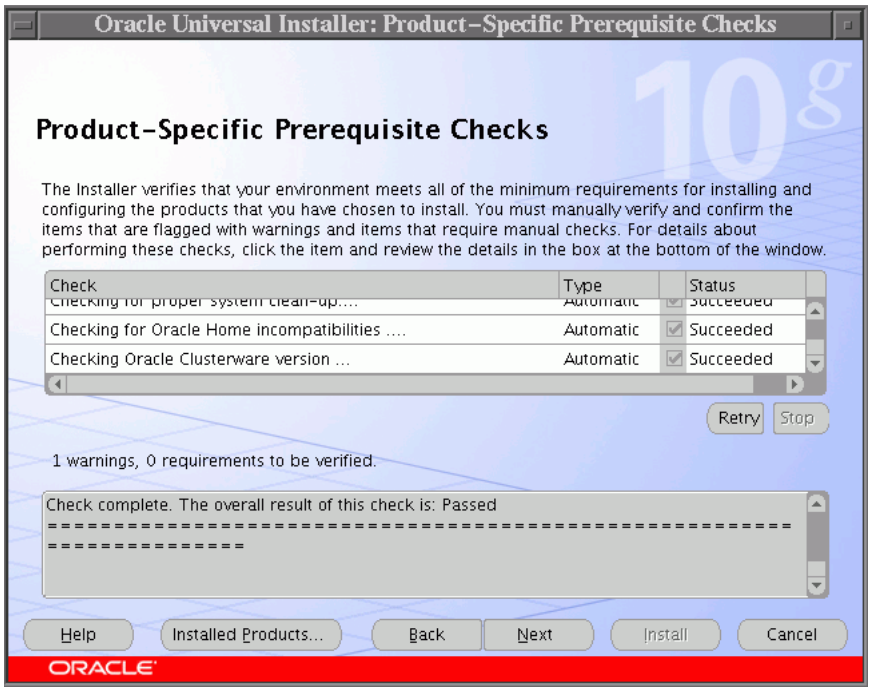
After you have specified the ASM Home directory, click **Next**.

The Specify Hardware Cluster Installation Mode window appears.

4. If your Oracle Clusterware installation was successful, then the Specify Hardware Cluster Installation Mode window lists the nodes that you identified for your cluster, such as `docrac1` and `docrac2`. Click **Select All** to select all nodes for installation, and then click **Next**.

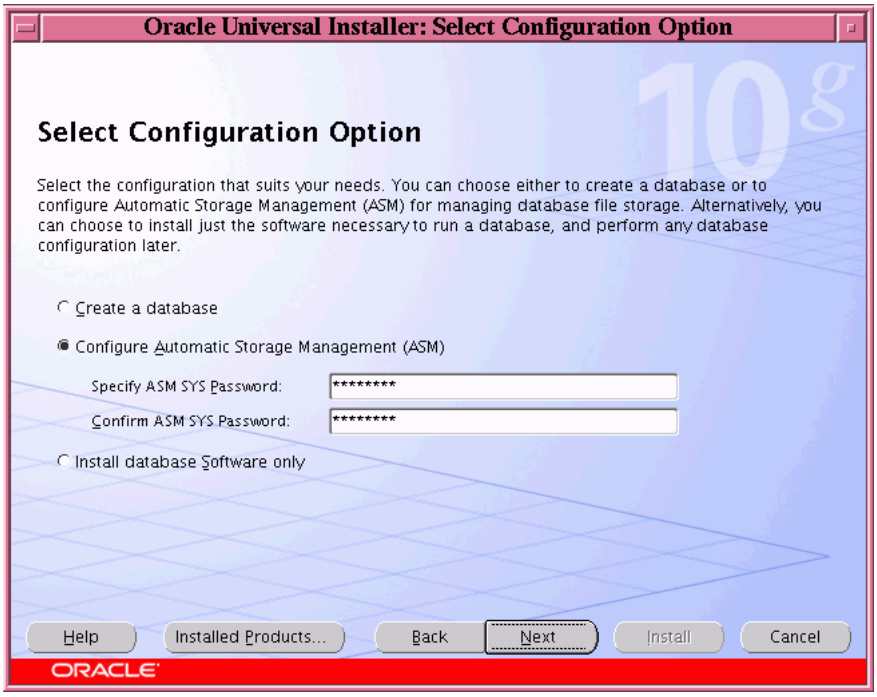
   The Product-Specific Prerequisites Checks window appears.

5. When you see the message "Check complete. The overall result of this check is: Passed", as shown in the following screenshot, click **Next**.



The Select Configuration Option window opens.

6. Select the **Configure Automatic Storage Management (ASM)** option to install and configure ASM. Enter a password for the ASM `sys` user. Confirm the password by typing it in again in the Confirm ASM SYS Password field. Then click **Next**.



The Configure Automatic Storage Management window appears.

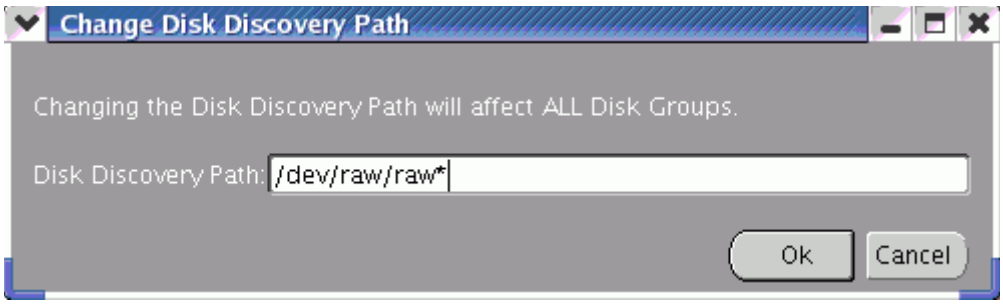
7. You configure ASM by creating disk groups that become the default location for files created in the database. The disk group type determines how ASM mirrors

files. When you create a disk group, indicate whether the disk group is a **normal redundancy** disk group (2-way mirroring for most files by default), or a **high redundancy** disk group (3-way mirroring), or an **external redundancy** disk group (no mirroring by ASM). Use an external redundancy disk group only if your storage system already provides mirroring at the hardware level, or if you have no need for redundant data. The default disk group type is normal redundancy.

In the Configure Automatic Storage Management window, the Disk Group Name defaults to `DATA`. Enter a new name for the disk group, such as `diskgroup1`. Check with your system administrator to determine if the disks used by ASM are mirrored at the storage level. If so, select **External** for the redundancy. If the disks are not mirrored at the storage level, then choose **Normal** for the redundancy.

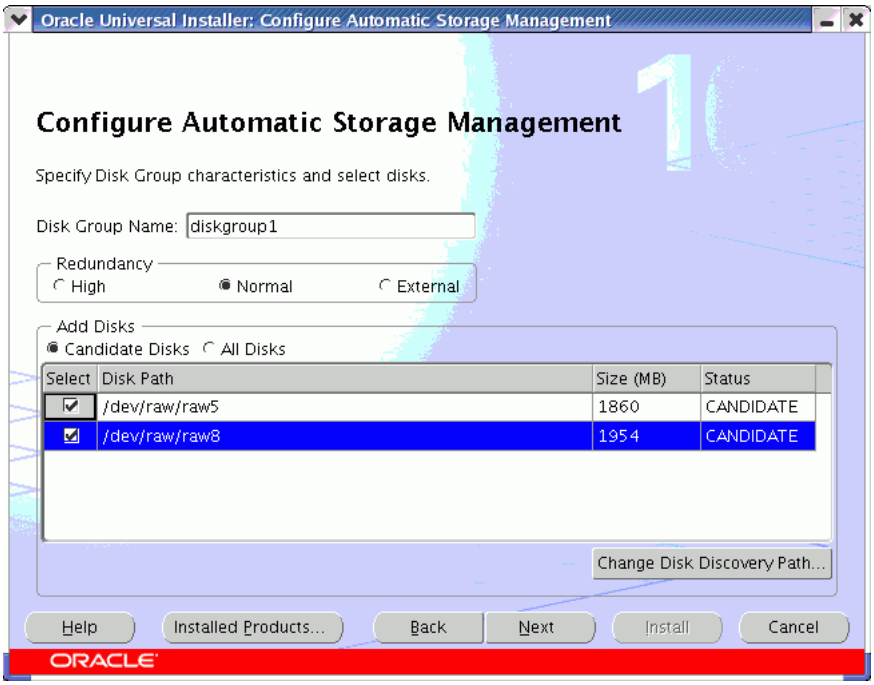
At the bottom right of the Add Disks section, click **Change Disk Discovery Path** to select any devices that will be used by ASM but are not listed.

In the Change Disk Discovery Path window, enter the path for the devices that ASM will use, such as `/dev/sd*` or `/dev/raw/raw*`. Then click **OK**.



You are returned to the Configure Automatic Storage Management window.

8. Select the disks to be used by ASM, for example, `/dev/raw/raw5` and `/dev/raw/raw8`. Then click **Next**.

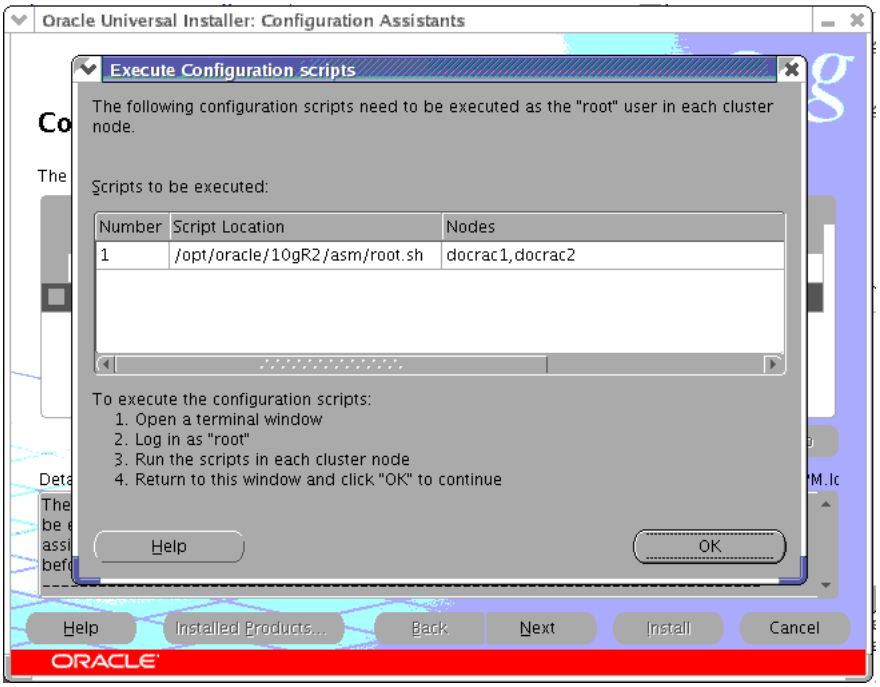


9. OUI displays the Summary window. Review the information displayed in this window. If any of the information appears incorrect, then you can click **Back** to return to a previous window and change it. When you are ready to proceed, click **Install**.

10. OUI displays a progress window indicating that the installation has started. The installation takes several minutes to complete. During this time, OUI configures ASM on the specified nodes, and then configures a Listener on those nodes.

    After ASM has been installed, OUI runs the Configuration Assistants. When the assistants have executed successfully, click the **Next** button to continue.

11. After the Configuration Assistants have completed their tasks, the Execute Configuration Scripts window appears. You are prompted to run one or more configuration scripts on the specified nodes.



You must run the scripts as instructed in the Execute Configuration scripts window before you click **OK**. For the installation demonstrated in this guide, only one script, `root.sh`, must be run, and it must be run on both nodes. The following steps demonstrate how to complete this task on a Linux system (note that for clarity, the examples show the user, node name, and directory in the prompt):

a. Open a terminal window. As the `oracle` user on `docrac1`, change directories to the ASM home directory, and then switch to the `root` user:

```
[oracle@docrac1 oracle]$ cd /opt/oracle/10gR2/asm
[oracle@docrac1 oracle]$ su
```

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

```
[root@docrac1 oracle]# ./root.sh
```

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. Press the **Enter** key each time you are prompted for input to accept the default choices.

d. After the script has completed, the prompt appears. Open another terminal window, and enter the following commands:

```
[oracle@docrac1 oracle]$ ssh docrac2
Enter the passphrase for key '/home/oracle/.ssh/id_rsa':
[oracle@docrac2 oracle]$ cd /opt/oracle/10gR2/asm
[oracle@docrac2 asm]$ su
Password:
```

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

```
[root@docrac2 asm]# ./root.sh
```

f. Accept all default choices by pressing Enter.

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

12. After you click **OK**, OUI displays the End of Installation window with Web addresses displayed. These Web addresses are not used in this guide. Click **Exit**, and then click **Yes** to verify that you want to exit the installation.

Verifying Your ASM Installation

Verify that all the database services for ASM are up and running. For example, on the `docrac1` node, change directories to the `bin` directory in the Oracle Clusterware home directory, and then run the following command as the `oracle` user:

```
cd /opt/oracle/crs/bin
./srvctl status asm -n docrac1
ASM instance +ASM1 is running on node docrac1.
```

The example output shows that there is one ASM instance running on the local node. Repeat the preceding command, substituting docrac2 for docrac1 to verify the successful installation on the other node in your cluster.

Installing the Oracle Database Software and Creating a Cluster Database

The next step is to install the Oracle Database 10*g* Release 2 software on the `docrac1` node. The installer copies the binary files from `docrac1` to `docrac2`, the other node in the cluster, during the installation process.

To install Oracle Database on your cluster:

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:

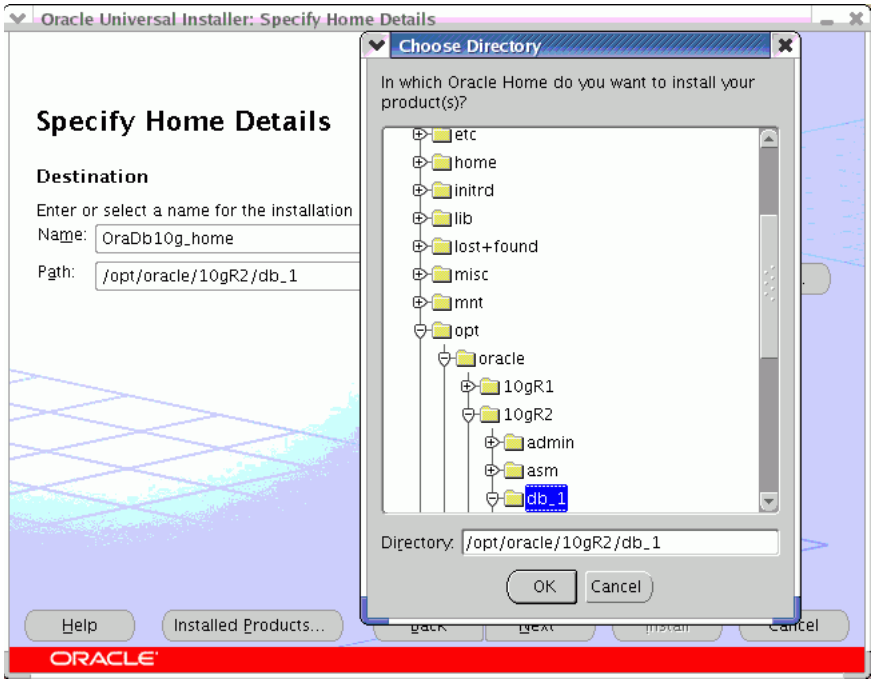
```
cd /staging_area/database
./runInstaller
```

The OUI Welcome window appears. Click **Next**.

2. The Select Installation Type window appears. The **Enterprise Edition** option is selected by default. Select either Enterprise Edition or Standard Edition and click **Next**.

3. The Specify Home Details window appears. Specify a name for the Oracle home, for example, `OraDb10g_home`. You must specify an Oracle home directory. Select a directory that is a subdirectory of your Oracle Base directory, for example, `/opt/oracle/10gR2/db_1`. Click **Browse** to change the directory in which the Oracle Database software will be installed. After you have selected the directory, click **OK**.

   If the directory does not exist, you can type in the directory path in the Directory field, then click **OK**. If a window appears asking if you want to create the directory, click **Yes**.



   When returned to the Specify Home Details window, verify the information is correct, then click **Next**.

   The Specify Hardware Cluster Installation Mode window appears.

4. Select the nodes on which the Oracle Database software will be installed. OUI is cluster-aware and hence knows the other nodes that are in the same cluster as the `docrac1` node.

   Because you are creating a cluster database, select both nodes by clicking **Select All**. Then click **Next**.

   The Product-Specific Prerequisite Checks window appears.

5. In this window, you might see a warning that says the host IP addresses are generated by the dynamic host configuration protocol (DHCP), which is not a recommended best practice. You can ignore this warning.

   When you see the confirmation message that your system has passed the prerequisite checks, click **Next**.

   The Select Configuration Option window appears.

6. In the Select Configuration Option window, accept the default option of **Create a Database** and click **Next**.

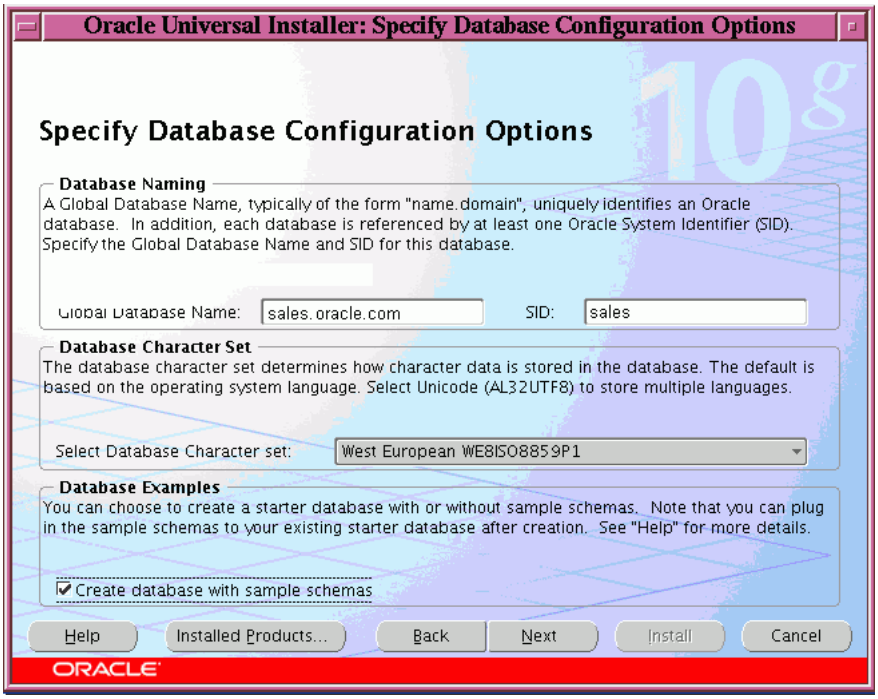
The Select Database Configuration window appears.

7. Choose one of the following different types of databases to be created:
   - General Purpose
   - Transaction Processing
   - Data Warehouse
   - Advanced (for customized database creation)

   The **General Purpose** database type is selected by default. Choose the type of database that best suits your business needs. For the example used by this guide, the default value is sufficient. After you have selected the database type, click **Next**.

   The Specify Database Configuration Options window appears.

8. Under Database Naming, in the Global Database Name field, enter a fully qualified name for your database, such as `sales.mycompany.com`. Ensure that the SID field contains the first part of the database name, for example, `sales`.



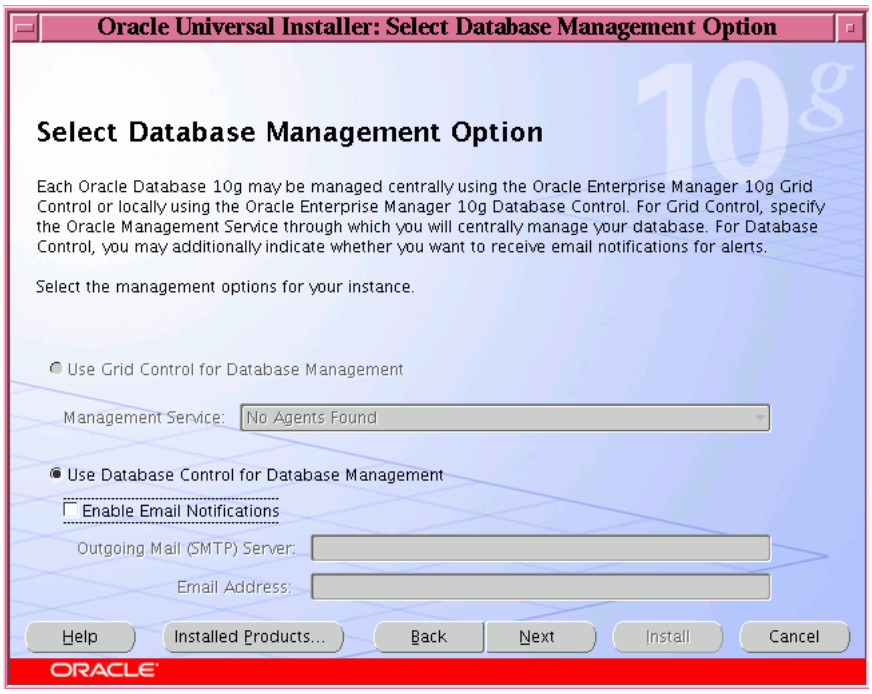
Note: The value for the SID will be used as a prefix for the instance names. Thus if the SID is set to `sales`, the instance names will be `sales1`, `sales2`, and so on.

Accept the default values for Database Character set (Western European WE8ISO8859P1) or specify a different language, as determined by your business requirements. Select the option **Create database with sample schemas** if you want sample data and schemas to be created in your database. After you have made your selections, click **Next**.

The Select Database Management Option window appears.

9. By default, the **Use Database Control for Database Management** option is selected instead of the Use Grid Control for Database Management option. The examples in this guide use Database Control, which is the default value.

   Under the option heading **Use Database Control for Database Management**, do not select the option Enable Email Notifications if your cluster is not connected to a mail server.



   After you have made your selections, click **Next**.

   The Specify Database Storage Option window appears.

10. If you configured ASM on the cluster, select the option **Automatic Storage Management (ASM)** for the database storage. Otherwise, select the option that you have decided upon for storing the database files, then click **Next**.

    The Specify Backup and Recovery Options window appears.

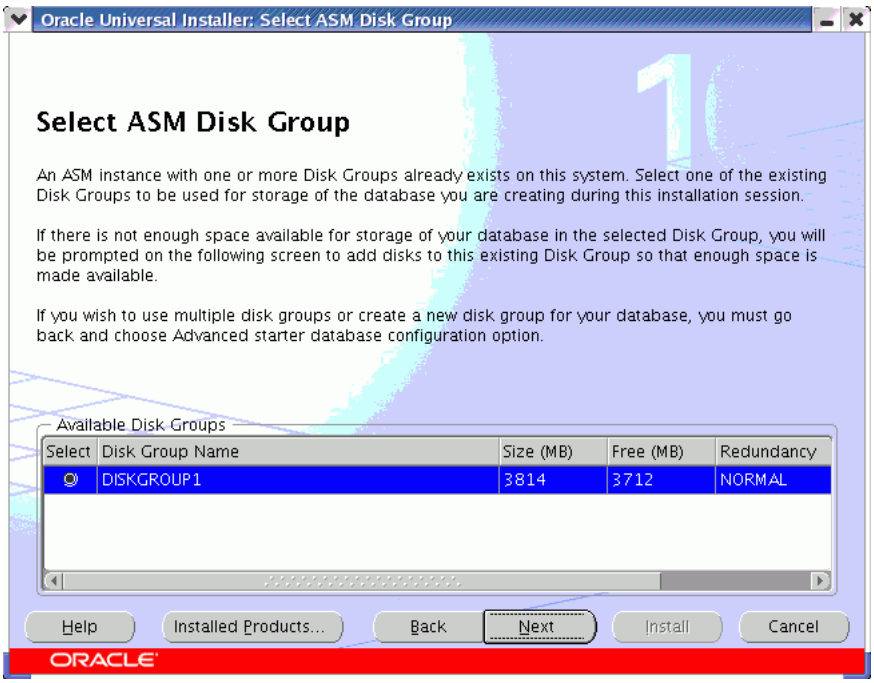
11. Select the default option **Do not enable Automated backup**, then click **Next**. You can modify the backup settings at a later time.

    If you chose ASM as your storage solution, the Select ASM Disk Group window appears.

    Note: If you want to use ASM as the backup area, you must create an additional ASM disk group when configuring ASM.

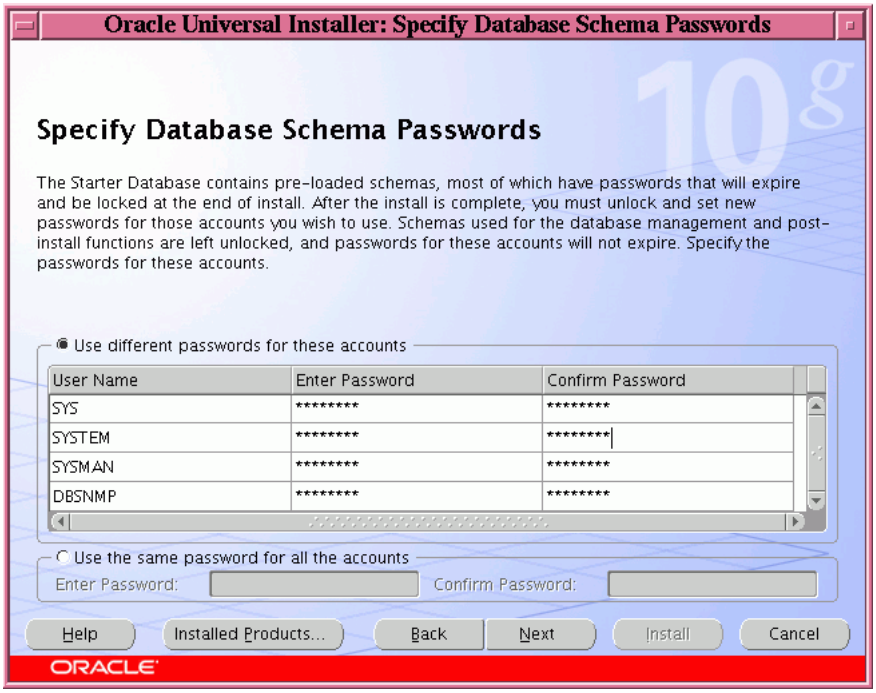
    See Also: *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information on configuring disk groups in ASM

12. The Select ASM Disk Group window shows you where the database files will be created. Select the disk group `diskgroup1` that was created during the ASM installation and then click **Next**.



The Specify Database Schema Passwords window appears.

13. Assign and confirm a password for each of the Oracle database schemas. Unless you are performing a database install action for testing purposes only, do *not* select the Use the same password for all the accounts option, as this can compromise the security of your data. When finished entering passwords, click **Next**.

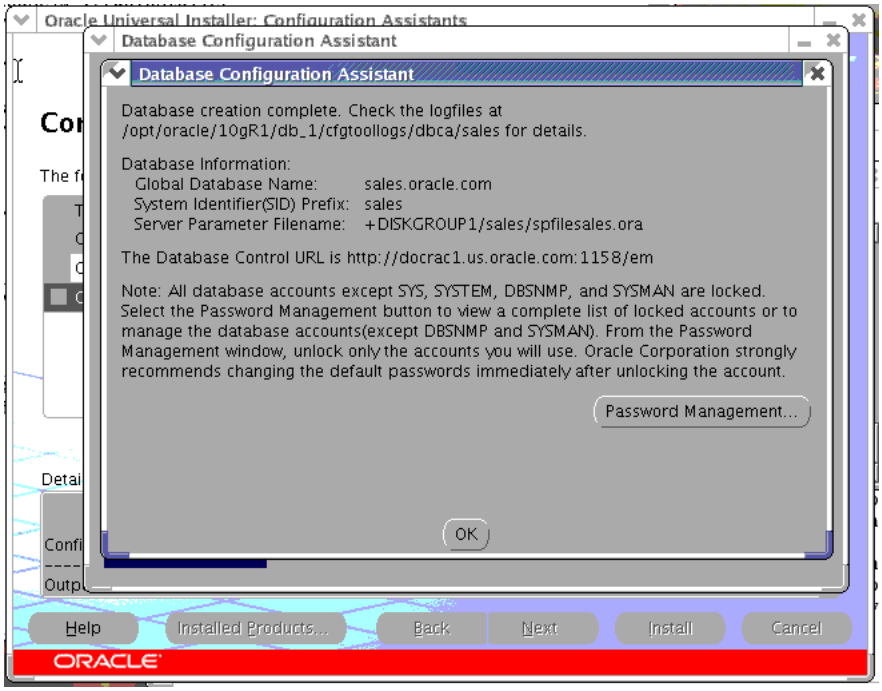


OUI displays the Summary window.

14. Review the information displayed in this window. If any of the information is incorrect, click **Back** to return to a previous window and correct it. When you are ready to proceed, click **Install**.

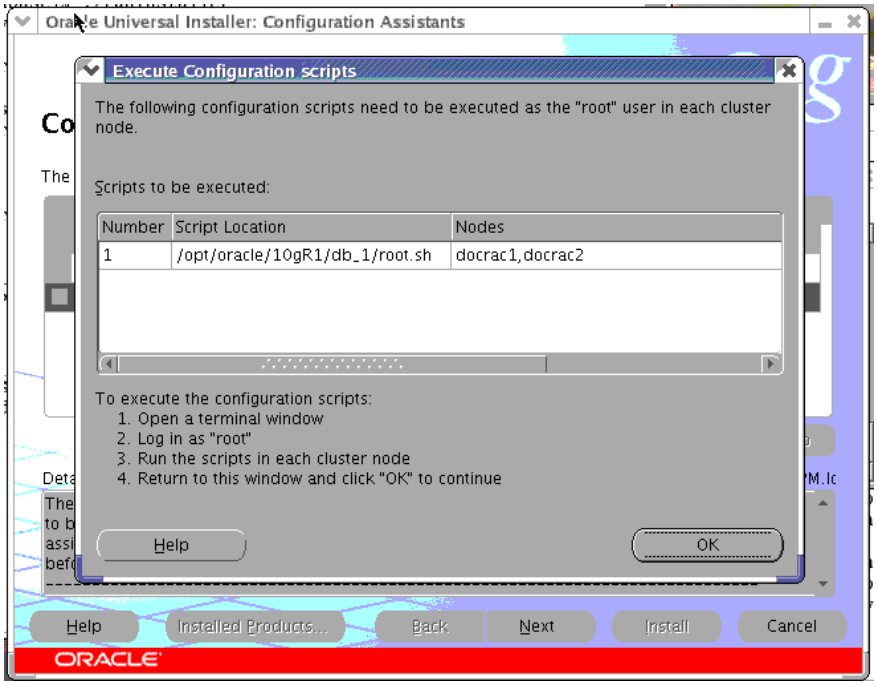
    OUI displays a progress indicator to show that the installation has begun. This step takes several minutes to complete.

15. As part of the software installation process, the `sales` database is created. At the end of the database creation, you will see the Database Configuration Assistant window with the Web address for the Database Control console displayed.



    Make note of the URL, then click **OK** and wait for DBCA to start the cluster database and its instances.

16. After the installation, you are prompted to perform the postinstallation task of running the `root.sh` script on both nodes.



On each node, run the scripts listed in the Execute Configuration scripts window before you click **OK**. Perform the following steps to run the `root.sh` script:

a. Open a terminal window. As the `oracle` user on `docrac1`, change directories to your Oracle home directory, and then switch to the `root` user by entering the following commands:

```
[oracle@docrac1 oracle]$ cd /opt/oracle/10gR2/db_1
[oracle@docrac1 db_1]$ su
```

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

```
[root@docrac1 db_1]# ./root.sh
```

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. Press the **Enter** key each time you are prompted for input to accept the default choices.

d. After the script has completed, the prompt appears. Open another terminal window, and enter the following commands:

```
[oracle@docrac1 oracle]$ ssh docrac2
[oracle@docrac2 oracle]$ cd /opt/oracle/10gR2/db_1
[oracle@docrac2 db_1]$ su
```

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

```
[root@docrac2 db_1]# ./root.sh
```

f. Accept all default choices by pressing the Enter key.

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

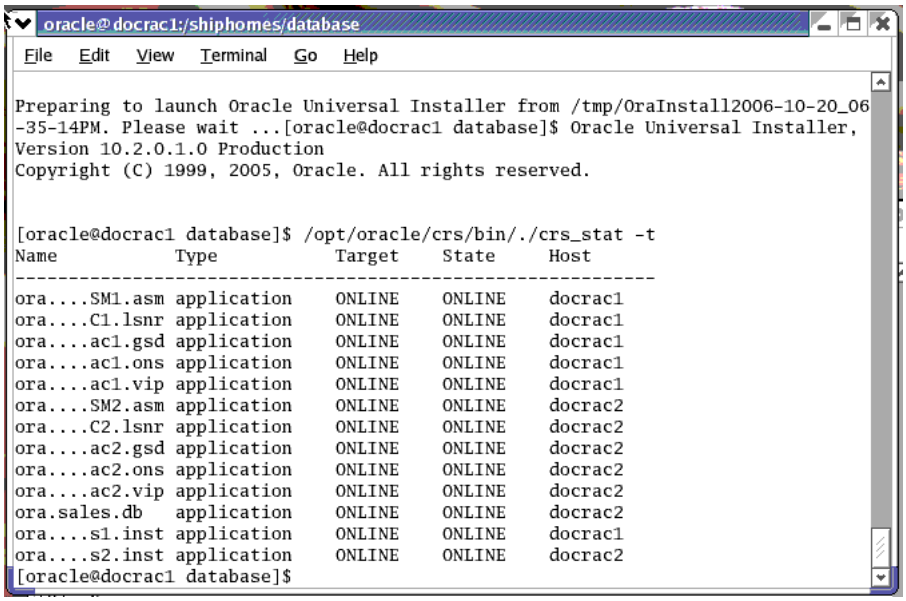
17. Click **OK** on the next window and OUI displays the End of Installation window. Click **Exit** and then click **Yes** to verify that you want to exit.

Verifying your Oracle RAC Database Installation

At this point, you should verify all the database services are up and running. To do this, log in as `oracle` on the `docrac1` node, and run the following commands:

```
[oracle] $ cd /opt/oracle/crs/bin
[oracle] $ ./crs_stat -t
```

The output of the command should show that database processes are available for each host.



Configuring the Operating System Environment for Database Management

After you have installed the Oracle RAC software and created a cluster database, there are two additional tasks to perform to configure your operating system environment for easier database management:

- Update the oratab File
- Reconfigure the User Shell Profile

Update the oratab File

Several of the Oracle Database utilities use the `oratab` file to determine the available Oracle homes and instances on each node. The `oratab` file is created by the `root.sh` script and is updated by the Database Configuration Assistant when creating or deleting a database.

The following is an example of the `oratab` file:

```
# This file is used by ORACLE utilities. It is created by root.sh
# and updated by the Database Configuration Assistant when creating
# a database.

# A colon, ':', is used as the field terminator. A new line terminates
# the entry. Lines beginning with a pound sign, '#', are comments.
```

```
#
# Entries are of the form:
# $ORACLE_SID:$ORACLE_HOME:<N|Y>:
#
# The first and second fields are the system identifier and home
# directory of the database respectively. The third field indicates
# to the dbstart utility that the database should, "Y", or should not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
#
+ASM1:/opt/oracle/10gR2/asm:N
sales:/opt/oracle/10gR2/db_1:N
sales1:/opt/oracle/10gR2/db_1:N
```

To update the oratab file on Red Hat Linux after creating an Oracle RAC database:

1. Open the `/etc/oratab` file for editing by using the following command on the `docrac1` node:

   ```
   vi /etc/oratab
   ```

2. Add the `SID` and `ORACLE_HOME` for the local instance to the end of the `/etc/oratab` file, for example:

   ```
   sales1:/opt/oracle/10gR2/db_1:N
   ```

3. Save the file and exit the vi editor.
4. Modify the `/etc/oratab` file on each node in the cluster, adding in the appropriate instance information.

> **Note:** In a single-instance database, setting the last field of each entry to N disables the automatic startup of a database when the server it runs on is restarted. For an Oracle RAC database, these fields are set to N because Oracle Clusterware starts the instances and processes, not the `dbstart` utility.

Reconfigure the User Shell Profile

There are several environment variables that can be used with Oracle Database. These variables can be set manually in your current operating system session, using shell commands such as `set` and `export`.

You can also have these variables set automatically when you log in as a specific operating system user. To do this, modify the Bourne, Bash, or Korn shell configuration file (for example `.profile` or `.login`) for that operating system user.

To modify the oracle user's profile for the bash shell on Red Hat Linux:

1. As the `oracle` user, open the user profile in the `/home/oracle` directory for editing using the following commands:

   ```
   [oracle] $ cd $HOME
   [oracle] $ vi .bash_profile
   ```

2. Modify the following lines in the file so they point to the location of the newly installed database:

```
export ORACLE_SID=sales
export ORACLE_BASE=/opt/oracle/10gR2
export ORACLE_HOME=/opt/oracle/10gR2/db_1
export PATH=$PATH:$ORACLE_HOME/bin
```

3. Read and execute the changes made to the `.bash_profile` file:

```
source .bash_profile
```

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide for Linux* for information about configuring environmental variables on Linux systems

Performing Postinstallation Tasks

After you have installed the Oracle RAC software, there are additional tasks that you can perform before your cluster database is ready for use. These steps are recommended, but are not required.

This section contains the following topics:

- Verifying the Clusterware Installation
- Backing Up the Voting Disk
- Downloading and Installing RDBMS Patches
- Verifying Oracle Enterprise Manager Operations
- Recommended Postinstallation Tasks

Verifying the Clusterware Installation

After the Oracle Clusterware installation is complete, OUI automatically runs the `cluvfy` utility as a Configuration Assistant to verify that the Clusterware installation has been completed successfully.

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

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide for Linux* for more information about using the CVU and resolving configuration problems

Backing Up the Voting Disk

After your Oracle Database 10*g* with Oracle RAC installation is complete and after you are sure that your system is functioning properly, make a backup of the contents of the voting disk. Use the `dd` utility, as described in the section "Backing Up and Recovering Voting Disks" in Chapter 5 of this guide.

Also, make a backup copy of the voting disk contents after you complete any node additions or deletions, and after running any deinstallation procedures.

Downloading and Installing RDBMS 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 only, with no upgrade or change to the database.

Ensure that you are running the latest patch set of the installed software. You might also need to apply 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".

Verifying Oracle Enterprise Manager Operations

When you install the Oracle RAC Database software and choose Database Control for your database management, the Enterprise Manager Database Control utility is installed and configured automatically.

To verify Oracle Enterprise Manager Database Control has been started in your new Oracle RAC environment:

1. Go to the `$ORACLE_HOME/bin` directory.
2. Run the following command as the `oracle` user:

   ```
   ./emctl status dbconsole
   ```

   The EMCTL utility displays the current status of the Database Control console on the current node.
3. If the EMCTL utility reports that Database Control is not started, use the following command to start it:

   ```
   ./emctl start dbconsole
   ```
4. Repeat steps 1 through 3 for each node in the cluster.

> **See Also:** *Oracle Database 2 Day DBA* for more information about managing the Enterprise Manager interface

Recommended Postinstallation Tasks

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

- Backing Up the root.sh Script
- Configuring User Accounts

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, 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.

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.

> **See Also:** *Oracle Database Administrator's Reference for UNIX-Based Operating Systems* for more information about setting up optional operating system user accounts that can be used to manage the database

Converting an Oracle Database to an Oracle RAC Database

You can optionally use Oracle Database Configuration Assistant (DBCA) to convert from a single-instance Oracle database to an Oracle RAC database. The DBCA automates the configuration of the control file attributes, creates the undo tablespaces and the redo logs, and makes the initialization parameter file entries for cluster-enabled environments. It also configures Oracle Net Services, Oracle Clusterware resources, and Oracle Enterprise Manager.

This section contains the following topics:

- Checking the Prerequisites
- Making a Preconfigured Copy of the Single-Instance Database
- Performing the Preinstallation Steps
- Validating the Cluster
- Copying the Preconfigured Database Files
- Installing the Oracle Database 10g Software with Real Application Clusters

Checking the Prerequisites

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

- The existing database and the target Oracle RAC database must be on the same release of Oracle Database 10*g* 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 version 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 will 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.

Note: Before using individual Oracle Database 10*g* database products or options, refer to the product documentation library, which is available in the DOC directory on the 10*g* Release 2 (10.2) installation media, or on the OTN Web site at `http://www.oracle.com/technology/documentation`

Making a Preconfigured Copy of the Single-Instance Database

As part of the database conversion process, you can use DBCA to create a preconfigured image of your database.

To create a preconfigured image of your single-instance database using DBCA:

1. Go to the `bin` directory in `$ORACLE_HOME`, and start DBCA.
2. At the Welcome window, click **Next**.
3. On the Operations window, select **Manage Templates**, and click **Next**.

4. On the Template Management window, select **Create a database template** and **From an existing database (structure as well as data)**, and click **Next**.
5. On the Source Database window, enter the database name in the Database instance field, and click **Next**.
6. On the Template Properties window, enter a name for your template in the Name field. Oracle recommends that you use the database name, for example, `sales`.

   By default, the template files are generated in the directory `$ORACLE_HOME/assistants/dbca/templates`. If you choose to do so, you can enter a description of the file in the Description field, and change the template file location in the Template datafile field.

   When you have finished entering the information, click **Next**.
7. On the Location of Database Related Files window, select **Maintain the file locations**, so that you can restore the database to the current directory structure, and click **Finish**.

DBCA generates two files: a database structure file (*`template_name`*`.dbc`), and a database preconfigured image file (*`template_name`*`.dfb`).

Performing the Preinstallation Steps

Follow the steps documented in Chapter 2 of this guide, titled "Preparing Your Cluster". You must do the following:

- Configure the servers to act as nodes in your cluster.
- Configure shared storage for the nodes in your cluster.
- Configure the interconnect and network connectivity between the nodes in your cluster.

Validating the Cluster

Validate the cluster configuration using the CVU, as described previously in this chapter in the section "Verifying the Configuration Using the Cluster Verification Utility".

Copying the Preconfigured Database Files

Copy the database structure `*.dbc` file and the database preconfigured image `*.dfb` files that DBCA created in the previous section titled "Making a Preconfigured Copy of the Single-Instance Database" to a temporary location on the node in the cluster from which you plan to run DBCA.

Installing the Oracle Database 10*g* Software with Real Application Clusters

After you have copied the preconfigured database files to the new node, install the Oracle RAC software on the new node. During the installation process, you will use the template files you created previously to convert your single-instance database to an Oracle RAC database.

To install the Oracle RAC software and convert your single-instance database to a cluster database:

1. Start OUI to perform an Oracle Database installation with Oracle RAC.

2. Select **Cluster Installation Mode** in the Specify Hardware Cluster Installation window of OUI, and select the nodes to include in your Oracle RAC database.
3. In the Database Configuration Types window, select the **Advanced** installation type.

   After installing the Oracle Database software, OUI runs postinstallation configuration tools, such as Network Configuration Assistant (NETCA), DBCA, and so on.
4. In the DBCA Template Selection window, use the template that you copied to a temporary location in the section "Copying the Preconfigured Database Files". Use the browse option to select the template location.
5. If you selected raw storage in the Storage Options window, then select the DBCA File Locations Tab on the Initialization Parameters window. Replace the datafiles, control files, and log files, and so on, with the corresponding raw device files. You must do this only if you have not set up the `DBCA_RAW_CONFIG` environment variable. You must also replace the default database files with raw devices in the Storage window.
6. After creating the Oracle RAC database, DBCA displays the Password Management window in which you must change the passwords for database-privileged users who have `SYSDBA` and `SYSOPER` roles.

When DBCA exits, the conversion process is complete.

4

Administering Database Instances and Cluster Databases

Web-based Oracle Enterprise Manager Database Control and Grid Control interfaces let you mange Oracle Real Application Clusters (Oracle RAC) databases. The Enterprise Manager console is a central point of control for the Oracle environment. Use the Database Control console to initiate cluster database management tasks. Use the Grid Control console to administer multiple Oracle RAC databases and cluster nodes.

This chapter describes how to administer your Oracle Real Application Clusters (Oracle RAC) environment. It explains the startup and shutdown tasks for database components and how to administer parameters and parameter files in Oracle RAC. This chapter includes the following sections:

- Overview of Oracle RAC Database Management
- Administering Oracle RAC with Enterprise Manager
- Starting and Stopping Oracle RAC Databases and Database Instances
- Viewing and Modifying Oracle RAC Initialization Parameters
- Administering Storage in Oracle RAC
- Exploring Your Cluster Database: Oracle By Example Series

Overview of Oracle RAC Database Management

Oracle RAC is a technology that links one or more individual computers so that they function as one system. Oracle RAC enables each computer that is a member of a cluster, or node, 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. This means that two or more inexpensive computers appear to applications as if they were a much more powerful, and more expensive, computer.

To increase the performance of a two-node 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 RAC database. The more nodes that your Oracle RAC environment has, the less the effect that the loss of any individual node has on the database.

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

See Also: Chapter 9, "Adding Nodes and Instances in Oracle Real Application Clusters Environments" for more information about adding nodes to your Oracle RAC cluster

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 Automatic Storage Management (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 some of the database administration tasks specific to Oracle RAC, as well as some recommendations for managing Oracle RAC databases.

Administering Oracle RAC with Enterprise Manager

The Web-based Oracle Enterprise Manager Database Control console and the Oracle Enterprise Manager Grid Control console let you manage Oracle RAC 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 Database Control for cluster database management tasks. Use Enterprise Manager Grid Control to administer your entire Oracle RAC environment, not just the Oracle RAC database.

Also note the following points about monitoring Oracle RAC environments:

- Performance monitoring features, such as Automatic Workload Repository and Statspack, are Oracle RAC-aware.

  Tip: Instead of using Statspack, Oracle recommends that you use the more sophisticated management and monitoring features of the Oracle Database *10g* Diagnostic and Tuning packs which include the Automatic Workload Repository (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.

When you log in to Enterprise Manager using a client browser, 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 database. This includes a summary about alert messages and job activity, as well as links to all the database and Automatic Storage Management (ASM) instances.

See Also: *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about monitoring Oracle RAC performance

Overview of Oracle Real Application Clusters: Oracle By Example Series

Oracle By Example (OBE) has a series of tutorials created for *Oracle Database 2 Day DBA*. Included in this series is an OBE tutorial that introduces you to the management of an Oracle RAC database using Enterprise Manager. To view this OBE tutorial, go to the following URL

http://www.oracle.com/technology/obe/10gr2_2day_dba/rac/rac.htm

Starting and Stopping Oracle RAC Databases and Database Instances

Typically, you start up 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. This enables you to more easily manage an Oracle RAC database.

You can also start and stop individual instances in an Oracle RAC database. However, starting and stopping one instance in an Oracle RAC database does not stop or start other 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. Point your Web browser to the following URL and log in to Enterprise Manager:

   `http://hostname:portnumber/em`

   For example, `http://docrac1.mycompany.com:1158/em`.

2. On the Cluster Database Home page, in the General section, click **Startup** if the database is down, or **Shutdown** if the database is started.
3. On the Startup/Shutdown: Specify Credentials page, enter the cluster database host credentials for the database 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.
4. On the Startup/Shutdown: Select Operation page, click **Startup All** to start all the instances, or click **Shutdown All** to stop all the instances.
5. On the Startup/Shutdown: Confirmation page, click **Yes**.

To start and stop individual instances, go to the Oracle RAC Database Startup and Shutdown page and select the instance that you want to start or stop. Then start or stop the instance as needed.

> **Note:** You can start up 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 Database Startup and Shutdown page.

You can also start up and shut down instances with SQL*Plus or Server Control (SRVCTL).

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

Viewing and Modifying Oracle RAC 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 number.

The administration of initialization parameters in Oracle RAC environments is slightly different from single-instance database parameter administration. For example, 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.

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

Configuring Initialization Parameters for an Oracle RAC 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* either be 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

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:

- `ACTIVE_INSTANCE_COUNT`
- `ARCHIVE_LAG_TARGET`
- `CLUSTER_DATABASE`

- CLUSTER_DATABASE_INSTANCES
- COMPATIBLE
- CONTROL_FILES
- DB_BLOCK_SIZE
- DB_DOMAIN
- DB_FILES
- DB_NAME
- DB_RECOVERY_FILE_DEST
- DB_RECOVERY_FILE_DEST_SIZE
- DB_UNIQUE_NAME
- INSTANCE_TYPE (RDBMS or ASM)
- PARALLEL_MAX_SERVERS
- REMOTE_LOGIN_PASSWORD_FILE
- UNDO_MANAGEMENT

The setting for DML_LOCKS must be identical on every instance only if set to zero.

Parameters that Must Have Unique Settings on All Instances

Oracle RAC uses the INSTANCE_NUMBER parameter to distinguish among instances at startup. Oracle RAC uses the number value of the THREAD parameter to assign redo log groups to specific instances. To simplify administration, use the same number for both the THREAD and INSTANCE_NUMBER parameters for each instance.

If you use the ROLLBACK_SEGMENTS parameter to specify the names of the rollback segments to be used for storing the undo of each instance, then Oracle recommends you use the instance SID as part of each unique rollback segment name. If the parameter UNDO_MANAGEMENT is set to AUTO, automatic undo management mode is used by the Oracle RAC database, and the setting of ROLLBACK_SEGMENTS is ignored. When using automatic undo management, Oracle Database generates unique names for the undo segments used by each instance.

If you use automatic undo management in your Oracle RAC database, then set the UNDO_TABLESPACE parameter to a different undo tablespace for each instance.

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:

- 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.

- LICENSE_MAX_USERS

This parameter determines a databasewide limit on the number of users defined in the database and 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 additional warning messages to be generated 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 archive log file names to have the format defined by its own value of `LOG_ARCHIVE_FORMAT`, regardless of which instance created the archive log files.

  Databases that support Oracle Data Guard, either to send or receive archive log files, must use the same value of `LOG_ARCHIVE_FORMAT` for all instances.

- `SPFILE`

  If this parameter does not identify the same file to all instances, then each instance may act differently and unpredictably in failover, load-balancing, or standard 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 will not be 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.

- `UNDO_RETENTION`

  By setting different values for `UNDO_RETENTION` in each instance, you are likely to reduce scalability and encounter unpredictable actions following a failover. Therefore, you should carefully consider whether or not you will accrue any benefits before you assign different values for this parameter to the instances in your Oracle RAC database.

Editing Initialization Parameter Settings for an Oracle RAC Database

To view or modify the initialization parameters with Enterprise Manager, click on the **Administration** tab on the Cluster Database home page. On the Administration page, click **Initialization Parameters** under Database Configuration. On the Initialization Parameters page, you can click either **Current** or **SPFile** to modify the parameter settings.

Modifying Initialization Parameter for Oracle RAC Using the Current Tab

Click the **Current** tab of the Initialization Parameters page in Enterprise Manager to manage the initialization parameter settings for your cluster database. The initialization parameters file 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.

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).

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 = 200`

for `docrac1` and `docrac2`, and `open_cursors = 275` for `docrac3`, then the Instance column for `open_cursors = 200` displays an asterisk, while displaying "docrac3" for `open_cursors = 275`. This shorthand saves space for cluster databases with many instances.

To modify the parameter values, select a parameter from the **Select** column and do one of the following:

- Click **Add** to add the selected parameter to a different instance. Enter a new instance name and value in the newly created row in the table.
- Click **Reset** to reset the value of the selected parameter. Note that you may only reset only parameters that do not have an asterisk in the Instance column. The value of the selected column is reset to the value of the remaining instances (that is, the row with the asterisk).

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

Modifying Initialization Parameter for Oracle RAC Using the SPFile Tab

From the Initialization Parameters page for cluster databases, select the **SPFile** tab to display and change the parameters for the current server parameter file. A server parameter file is a type of repository for initialization parameters that is maintained on the server where the Oracle database server runs. It is a server-side initialization parameter file. 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. Click **Apply changes in SPFile mode...** to apply the changes to the running instance.

Select **Running** to show the parameters for the current running instance of the initialization parameters file. For more information about each parameter, click the parameter name in the Name column.

Similar to the Current tab, you can Add or Reset parameters. Note that resetting parameters in the SPFile tab is different than resetting them in the Current tab: Reset deletes the selected parameter entry from the SPFILE and applies to both asterisk and non-asterisk parameters. If you reset a parameter with an asterisk in the Instance column, the entry will be deleted from both the SPFILE and the table. Only parameters for non-asterisk instances will remain. If you reset the remaining entry for a parameter, it will be deleted from both the SPFILE and the table, but will be replaced by a *dummy* parameter with an empty value field and an asterisk in the Instance column; this enables you to specify a new value, add new entries, and so on.

Resetting a parameter with only one instance resets the value of that parameter.

Example: Modifying the OPEN_CURSORS Parameter

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

```
*.open_cursors = 200
docrac1.open_cursors = 250
```

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 `docrac1.open_cursors = 250` displayed. If you click **Reset** for `docrac1.open_cursors`, Enterprise Manager also deletes this parameter entry from both the SPFILE and the displayed list of parameters, but then a new entry, `*.open_cursors = <EMPTY>` is added to the displayed list of parameters in place of the reset parameter.

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about using a server parameter file in an Oracle Real Application Clusters environment

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 Enterprise Manager. If you need to change a service, you can use either Enterprise Manager or SRVCTL.

> **See Also:** Chapter 7, "Managing Database Workload Using Services" for more information about using services with Oracle RAC

Configuring 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 ASM disk group, cluster file system file, or a shared raw device. In the environment described by this guide, the SPFILE is created on an ASM disk group.

All instances in the cluster database use the same SPFILE at startup. Oracle RAC uses a traditional PFILE only if an SPFILE does not exist or if you specify `PFILE` in your `STARTUP` command. Oracle recommends that you use 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.

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 instructions for using Enterprise Manager to manage some of the storage structures of an Oracle RAC database.

This section describes the following topics:

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

Administering Automatic Undo Management in Oracle RAC

Oracle 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, as long as 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

Administering Automatic Storage Management in Oracle RAC

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

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

- About Automatic Storage Management Components in Oracle RAC
- Modifying Disk Group Configurations for ASM in Oracle RAC
- Standalone ASM Disk Group Management
- Administering ASM Instances and Disk Groups with Enterprise Manager

About Automatic Storage Management Components in Oracle RAC

When you create your database, Oracle Database creates one ASM instance on each node in your Oracle RAC environment if one does not already exist. Each ASM instance has either an SPFILE or PFILE type parameter file. For the environment described in this guide, the ASM instances use PFILEs.

Modifying Disk Group Configurations for 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 ASM in an Oracle RAC database compared to using it in a single-instance Oracle database. ASM automatically rebalances the storage load after you add or delete a disk or disk group.

In a cluster, each ASM instance manages the metadata updates to the disk groups for the node on which it is running. In addition, each ASM instance coordinates disk group metadata with other nodes in the cluster. As in single-instance Oracle databases, you can use Enterprise Manager, DBCA, SQL*Plus, and SRVCTL to administer disk groups for ASM in an Oracle RAC environment.

> **See Also:** *Oracle Database Administrator's Guide* for information on how to use SQL*Plus to administer ASM instances

Standalone ASM Disk Group Management

When you create a database using DBCA and you select the ASM storage option, DBCA creates the ASM instances for you if they do not already exist. However, you can also use the standalone ASM disk group management feature to create and

manage an ASM instance and its associated disk groups independently of creating a new database.

Administering ASM Instances and Disk Groups with Enterprise Manager

You can perform administrative operations on ASM disk groups with Enterprise Manager such as adding and deleting them. You can also monitor ASM disk group performance as well as control disk group availability at the instance level. For example, some of the Oracle RAC-specific features for ASM that are provided by Enterprise Manager are:

- When you add a disk group, the disk group definition includes a check box to indicate whether or not 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 ASM disk groups, you can use a check box to indicate which instances should mount or dismount a particular ASM Disk Group.

See Also:

- *Oracle Database Administrator's Guide*
- *Oracle Database 2 Day DBA*

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. This section provides an overview of some of the additional concepts and procedures for configuring redo log files in Oracle RAC environments.

About Redo Log Groups and Redo Threads in Oracle RAC Databases

Redo logs contain a record of changes that have been made to datafiles. 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 the Oracle Database attempts to access its mirrored copy. You should place the redo log file mirrors on different disk devices from the primary redo log files.

Each instance's redo thread must contain at least two redo log groups. Oracle recommends that each of your instances has a redo thread that contains the same number of redo log groups and, as with single-instance Oracle databases, each group should contain the same number of members. For example, in an Oracle RAC database with two instances, each instance could have a redo thread that contains five redo log groups. This is a total of 10 redo log groups for the database. Each of these redo log groups could contain two members: a redo log and its mirrored copy. If you create your Oracle RAC database with DBCA, then your Oracle RAC database automatically implements a configuration that meets the Oracle recommendations.

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 is able to read and process all

the redo threads for the database, regardless of which instance generated the redo thread. This enables a running instance to recover the work completed by one or more failed instances. This also enables users to continue their work 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.

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 ASM, then ASM manages the shared storage for the redo log files and the access to those files.

Using Enterprise Manager to View and Create Online Redo Log Files

To access the redo log file groups with Enterprise Manager:

1. Go to the Cluster Database Home Page, and click the **Administration** tab.
2. On the Cluster Database Administration page, under the Storage column, select **Redo Log Groups**.

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.

See Also:

- *Oracle Database 2 Day DBA* for more information about creating online redo log files
- *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for additional information about redo threads in an Oracle RAC environment

Exploring Your Cluster Database: Oracle By Example Series

Oracle By Example (OBE) has a series of tutorials for Oracle RAC databases. This OBE steps you through the basic administrative tasks described in this chapter and includes annotated screen shots.

To view the Exploring Your Cluster Database OBE, go to the Web site

http://www.oracle.com/technology/obe/10gr2_db_vmware/manage/clusterintro/clusterintro.htm

5

Administering Oracle Clusterware Components

This chapter describes how to administer your Oracle Clusterware environment. It describes how to administer the voting disks and the Oracle Cluster Registry (OCR) in the following sections:

- About Oracle Clusterware
- Backing Up and Recovering Voting Disks
- Adding and Removing Voting Disks
- Backing Up and Recovering the Oracle Cluster Registry
- Changing the Oracle Cluster Registry Configuration
- Troubleshooting the Oracle Cluster Registry

About Oracle Clusterware

Oracle Real Application Clusters (Oracle RAC) uses Oracle Clusterware as the infrastructure that binds together multiple nodes that then operate as a single server. Oracle Clusterware is a portable cluster management solution that is integrated with Oracle Database. In an Oracle RAC environment, Oracle Clusterware monitors all Oracle components (such as instances and Listeners). If a failure occurs, 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 make sure that 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.

Oracle Clusterware includes two important components: the voting disk and the OCR. The voting disk is a file that manages information about node membership, and the OCR is a file that manages cluster and Oracle RAC database configuration information.

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.

Backing Up and Recovering Voting Disks

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 backup component.

The voting disk records node membership information. A node must be able to access more than half of 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 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.

Backing Up Voting Disks

Because the node membership information does not usually change, you do not need to back up the voting disk every day. However, back up the voting disks at the following times:

- After installation
- After adding nodes to or deleting nodes from the cluster
- After performing voting disk `add` or `delete` operations

To make a backup copy of the voting disk, use the Linux `dd` command. Perform this operation on every voting disk as needed where *`voting_disk_name`* is the name of the active voting disk and *`backup_file_name`* is the name of the file to which you want to back up the voting disk contents:

```
dd if=voting_disk_name of=backup_file_name
```

If your voting disk is stored on a raw device, use the device name in place of *`voting_disk_name`*. For example:

```
dd if=/dev/sdd1 of=/tmp/voting.dmp
```

When you use the `dd` command for making backups of the voting disk, the backup can be performed while the Cluster Ready Services (CRS) process is active; you do not need to stop the `crsd.bin` process before taking a backup of the voting disk.

Recovering Voting Disks

If a voting disk is damaged, and no longer usable by Oracle Clusterware, you can recover the voting disk if you have a backup file. Run the following command to recover a voting disk where *`backup_file_name`* is the name of the voting disk backup file and *`voting_disk_name`* is the name of the active voting disk:

```
dd if=backup_file_name of=voting_disk_name
```

Adding and Removing Voting Disks

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. Run the following command as the `root` user to add a voting disk:

```
crsctl add css votedisk path
```

Run the following command as the `root` user to remove a voting disk:

```
crsctl delete css votedisk path
```

Note: If your cluster is down, then you can use the `-force` option to modify the voting disk configuration when using either of these commands without interacting with active Oracle Clusterware daemons. However, you may corrupt your cluster configuration if you use the `-force` option while a cluster node is active.

Backing Up and Recovering the Oracle Cluster Registry

Oracle Clusterware automatically creates OCR backups every 4 hours. At any one time, Oracle Clusterware always retains the latest 3 backup copies of the OCR that are 4 hours old, 1 day old, and 1 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. The default location for generating backups on Red Hat Linux systems is *CRS_home*/cdata/*cluster_name* where *cluster_name* is the name of your cluster and *CRS_home* is the home directory of your Oracle Clusterware installation.

This section contains the following topics:

- Viewing Available OCR Backups
- Backing Up the OCR
- Recovering the OCR

Viewing Available OCR Backups

To find the most recent backup of the OCR, on any node in the cluster, use the following command:

```
ocrconfig -showbackup
```

Backing Up the OCR

Because of the importance of OCR information, Oracle recommends that you use the `ocrconfig` tool to make copies of the automatically created backup files at least once a day.

In addition to using the automatically created OCR backup files, you should also export the OCR contents to a file before and after making significant configuration changes, such as adding or deleting nodes from your environment, modifying Oracle

Clusterware resources, or creating a database. Exporting the OCR contents to a file lets you restore the OCR if your configuration changes cause errors. For example, if you have unresolvable configuration problems, or if you are unable to restart your cluster database after such changes, then you can restore your configuration by importing the saved OCR content from the valid configuration.

To export the contents of the OCR to a file, use the following command, where *backup_file_name* is the name of the OCR backup file you want to create:

```
ocrconfig -export backup_file_name
```

Note: You must be logged in as the `root` user to run the `ocrconfig` command.

Recovering the OCR

This section describes 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.

In event of a failure, before you attempt to restore the OCR, ensure that the OCR is unavailable. Run the following command to check the status of the OCR:

```
ocrcheck
```

If this command does not display the message `'Device/File integrity check succeeded'` for at least one copy of the OCR, then both the primary OCR and the OCR mirror have failed. You must restore the OCR from a backup.

Restoring the Oracle Cluster Registry from Automatically Generated OCR Backups

When restoring the OCR from automatically generated backups, you first have to determine which backup file you will use for the recovery.

To restore the OCR from an automatically generated backup on a Red Hat Linux system:

1. Identify the available OCR backups using the `ocrconfig` command:

   ```
   # ocrconfig -showbackup
   ```

   Note: You must be logged in as the `root` user to run the `ocrconfig` command.

   ---

2. Review the contents of the backup using the following `ocrdump` command, where *file_name* is the name of the OCR backup file:

   ```
   $ ocrdump -backupfile file_name
   ```

3. As the `root` user, stop Oracle Clusterware on all the nodes in your Oracle RAC cluster by executing the following command:

   ```
   # crsctl stop crs
   ```

   Repeat this command on each node in your Oracle RAC cluster.

4. As the `root` user, restore the OCR by applying an OCR backup file that you identified in step 1 using the following command, where *file_name* is the name

of the OCR that you want to restore. Make sure that the OCR devices that you specify in the OCR configuration exist, and that these OCR devices are valid before running this command.

```
# ocrconfig -restore file_name
```

5. As the `root` user, restart Oracle Clusterware on all the nodes in your cluster by restarting each node, or by running the following command:

   ```
   # crsctl start crs
   ```

   Repeat this command on each node in your Oracle RAC cluster.

6. Use the Cluster Verify Utility (CVU) to verify the OCR integrity. 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]
   ```

Recovering the OCR from an OCR Export File

Using the `ocrconfig -export` command enables you to restore the OCR using the `-import` option if your configuration changes cause errors.

To restore the previous configuration stored in the OCR from an OCR export file:

1. Place the OCR export file that you created previously with the `ocrconfig -export` command in an accessible directory on disk.
2. As the `root` user, stop Oracle Clusterware on all the nodes in your Oracle RAC cluster by executing the following command:

   ```
   crsctl stop crs
   ```

   Repeat this command on each node in your Oracle RAC cluster.

3. As the `root` user, restore the OCR data by importing the contents of the OCR export file using the following command, where *`file_name`* is the name of the OCR export file:

   ```
   ocrconfig -import file_name
   ```

4. As the `root` user, restart Oracle Clusterware on all the nodes in your cluster by restarting each node, or by running the following command:

   ```
   crsctl start crs
   ```

   Repeat this command on each node in your Oracle RAC cluster.

5. Use the CVU to verify the OCR integrity. 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]
   ```

Note: You cannot use the `ocrconfig` command to import an OCR backup file.

Changing the Oracle Cluster Registry Configuration

This section describes how to administer the OCR. The OCR contains information about the cluster node list, which instances are running 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
- Replacing an OCR
- Repairing an Oracle Cluster Registry Configuration on a Local Node
- Removing an Oracle Cluster Registry

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. So, you should avoid shutting down nodes while modifying the OCR using the `ocrconfig` command.

Adding an OCR Location

You can add an OCR location after an upgrade or after completing the Oracle RAC installation. If you already mirror the OCR, then you do not need to add an OCR location; Oracle Clusterware automatically manages two OCRs when you configure normal redundancy for the OCR. Oracle RAC environments do not support more than two OCRs, a primary OCR and a secondary OCR.

Run the following command to add an OCR location using either *`destination_file`* or *`disk`* to designate the target location of the additional OCR:

```
ocrconfig -replace ocr destination_file
ocrconfig -replace ocr disk
```

Run the following command to add an OCR mirror location using either *`destination_file`* or *`disk`* to designate the target location of the additional OCR:

```
ocrconfig -replace ocrmirror destination_file
ocrconfig -replace ocrmirror disk
```

Note: You must be logged in as the `root` user to run the `ocrconfig` command.

Replacing an OCR

If you need to change the location of an existing OCR, or change the location of a failed OCR to the location of a working one, you can use the following procedure as long as one OCR file remains online.

To change the location of an OCR:

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. Verify that Oracle Clusterware is running on the node on which the you are going to perform the replace operation using the following command:

   ```
   crsctl check crs
   ```

3. Run the following command to replace the OCR using either *destination_file* or *disk* to indicate the target OCR:

   ```
   ocrconfig -replace ocr destination_file
   ocrconfig -replace ocr disk
   ```

4. Run the following command to replace an OCR mirror location using either *destination_file* or *disk* to indicate the target OCR:

   ```
   ocrconfig -replace ocrmirror destination_file
   ocrconfig -replace ocrmirror disk
   ```

5. If any node that is part of your current Oracle RAC environment is shut down, then run the following command on the stopped node to let that node rejoin the cluster after the node is restarted:

   ```
   ocrconfig -repair
   ```

Repairing an Oracle Cluster Registry Configuration on a Local Node

You may need to repair an OCR configuration on a particular node if your OCR configuration changes while that node is stopped. For example, you may need to repair the OCR on a node that was shut down while you were adding, replacing, or removing an OCR. To repair an OCR configuration, run the following command on the node on which you have stopped the Oracle Clusterware daemon:

```
ocrconfig -repair ocrmirror device_name
```

> **Note:** You *cannot* perform this operation on a node on which the Oracle Clusterware daemon is running.

This operation changes the OCR configuration only on the node from which you run this command. For example, if the OCR mirror is on a disk named `/dev/raw1`, then use the command `ocrconfig -repair ocrmirror /dev/raw1` on this node to repair its OCR configuration.

Removing an Oracle Cluster Registry

To remove an OCR location, at least one OCR must be online. You can remove an OCR location to reduce OCR-related overhead or to stop mirroring your OCR because you moved your the OCR to a redundant storage system, such as a redundant array of independent disks (RAID).

To remove an OCR location from your Oracle RAC environment:

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. Run the following command on any node in the cluster to remove one copy of the OCR:

   ```
   ocrconfig -replace ocr
   ```

   This command updates the OCR configuration on all the nodes on which Oracle Clusterware is running.

Troubleshooting the Oracle Cluster Registry

This section includes the following topics on how to troubleshoot the Oracle Cluster Registry (OCR):

- Using the OCRCHECK Utility
- Resolving Common Oracle Cluster Registry Problems

Using the OCRCHECK Utility

The OCRCHECK utility displays the data block format version used by the OCR, the free 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 as well as 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                  :          2
   Total space (kbytes)     :     262144
   Used space (kbytes)      :      16256
   Available space (kbytes) :     245888
   ID                       : 1918913332
   Device/File Name         : /dev/raw/raw1
                              Device/File integrity check succeeded
   Device/File Name         : /oradata/mirror.ocr
                              Device/File integrity check succeeded

Cluster registry integrity check succeeded
```

The OCRCHECK utility creates a log file in the following directory, where *CRS_home* is the location of the installed Oracle Clusterware software, and *hostname* is the name of the local node:

```
CRS_home/log/hostname/client
```

The log files have names of the form `orcheck_nnnnn.log`, where *nnnnn* is the process ID of the operating session that issued the `ocrcheck` command.

Resolving Common Oracle Cluster Registry Problems

Table 5–1 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 `-replace` option as described in the section "Replacing an OCR" on page 5-6.
An OCR mirror 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 "Adding an OCR Location" on page 5-6.
An OCR has been incorrectly updated.	Run the `ocrconfig` command with the `-repair` option as described in the section "Repairing an Oracle Cluster Registry Configuration on a Local Node" on page 5-7.
You are experiencing a severe performance effect from OCR processing, or you want to remove an OCR for other reasons.	Run the `ocrconfig` command with the `-repair` option as described in the section "Repairing an Oracle Cluster Registry Configuration on a Local Node" on page 5-7.

6

Administering Backup and Recovery

This chapter describes how to back up and recover an Oracle Real Application Clusters (Oracle RAC) database.

This chapter contains the following sections:

- Overview of Oracle RAC Database Backup and Recovery
- Deploying a Flash Recovery Area in Oracle Real Application Clusters
- Archiving in Oracle Real Application Clusters
- Credentials for Performing Enterprise Manager Backup and Recovery
- Performing Backups of Your Oracle RAC Database
- Preparing to Restore and Recover Your Oracle RAC Database
- Recovering Your Oracle RAC Database
- Managing Your Database Backups
- Displaying Backup Reports for Your Oracle RAC Database
- Performing Backups and Recovering Your Database: Oracle By Example Series

> **See Also:** *Oracle Database Backup and Recovery Basics* for more information about using the Recovery Manager utility

Overview of Oracle RAC Database Backup and Recovery

To protect your Oracle RAC database from hardware failures or disasters, you need to have a physical copy of the database files. The files protected by the backup and recovery facilities built into Oracle Enterprise Manager include datafiles, control files, server parameter files (SPFILEs), and archived redo log files. With 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 datafile or the failure of a disk drive. The process of restoring damaged files for your database is called **database recovery**.

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 oepations. The flashback features enable you to reverse the effects of unwanted database changes without restoring datafiles 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.

The Enterprise Manager Guided Recovery capability provides a Recovery wizard that encapsulates the logic required for a wide range of restore and recovery scenarios, including the following:

- Complete restore and recovery of the database
- Point-in-time recovery of the database or selected tablespaces
- Flashback Database
- Other flashback features of Oracle for logical-level repair of unwanted changes to database objects
- Media recovery at the block level for datafiles with corrupt blocks

Enterprise Manager can determine which parts of the database must be restored and recovered, including proactively detecting situations such as corrupted database files. Enterprise Managers walks you through the recovery process, prompting for needed information and performing required recovery actions.

> **See Also:** *Oracle Database 2 Day DBA* for more information about database backup, database recovery, and Oracle Flashback concepts

Deploying a Flash Recovery Area in Oracle Real Application Clusters

Using a flash 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 flash recovery area to simplify your backup management.

The larger the flash recovery area is, the more useful it becomes. Ideally, the flash recovery area should be large enough to contain all the following files:

- A copy of all datafiles
- Incremental backups
- Online redo logs
- Archived redo logs that have not yet been backed up
- Control files and control file copies
- Autobackups of the control file and database initialization parameter file

The preferred configuration for Oracle RAC is to use Automatic Storage Management (ASM) for a recovery area with a different disk group for your recovery set than for your datafiles. Alternatively, you can use a cluster file system archiving scheme.

The location and disk quota must be the same on all instances. To accomplish this, Oracle recommends that you place the flash recovery area on the shared 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 Flash Recovery feature, you must first configure the flash recovery area, as described in *Oracle Database 2 Day DBA*, for each instance in your Oracle RAC cluster.

Archiving in Oracle Real Application Clusters

When you archive your redo log, you write redo log files to another location prior to their being overwritten. This location is called the archive 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.

When you use the Database Configuration Assistant to create your Oracle RAC database, each instance is configured with at least two redo log files that are stored in the shared storage. If you use a cluster file system, then these files are shared file system files. If you do not have a cluster file system, then these files are raw devices. If you use ASM, then these files are stored on the ASM disk group.

Configuring Archiving for Your Oracle RAC Database

For Oracle Real Application Clusters, each instance has its own thread of redo. The preferred configuration for Oracle RAC is to configure the flash recovery area using an ASM disk group that is separate from the ASM disk group used for your datafiles. Alternatively, you can use a cluster file system archiving scheme.

If you use a cluster file system, you must specify `LOG_ARCHIVE_DEST_1` and `LOG_ARCHIVE_FORMAT` for each node in the database parameter initialization file.

See Also:

- *Oracle Database 2 Day DBA* for details on how to configure archiving using ASM
- *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring and managing archived redo logs for an Oracle RAC database

About Instance Access to Archived Log Files

An instance does not need access to the archived logs from a different instance except when performing backup or recovery operations. When performing backup operations across instances, the archived redo log naming scheme that you use is important because when an instance writes to a log with a specific file name on its file system, that file must be readable by any instance that needs to access this archived redo log.

Also, the backup and recovery strategy that you implement for your Oracle RAC database depends on how you configure the archiving destinations for each instance.

If you use ASM to store the archived redo logs for your Oracle RAC database, then each instance automatically has access to all the archived log files generated by the database. If you use shared storage or raw devices to store the archived log files on each node, then you must configure the operating system to grant access to those directories for each instance in the cluster database that needs access to them.

Credentials for Performing Enterprise Manager Backup and Recovery

You must have the proper credentials to perform some of the configuration tasks for backup and recovery, and to schedule backup jobs and 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 a check box labeled Save as Preferred Credential. If you check this box 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.

Configuring Backup Settings

Assuming you have a flash recovery area configured and are running in `ARCHIVELOG` mode, you can configure a number of 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 flash recovery area. You can also configure settings to optimize backup performance for your environment.

> **See Also:** *Oracle Database 2 Day DBA* for more information about using Enterprise Manager to configure backup settings for your database

Performing Backups of Your Oracle RAC Database

When you use 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, any operating system user who has execute permission for the RMAN command-line client can perform backups of an 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. Go to the Cluster Database Home Page, and click the **Maintenance** tab.
2. On the Cluster Database Maintenance page, under the Backup/Recovery column, select **Schedule Backup**.
3. Follow the backup procedures outlined in Chapter 9, "Performing Backup and Recovery" of *Oracle Database 2 Day DBA*.

About Parallelism and Backups Across Multiple Channels

RMAN depends on server sessions, processes that run on the database server, to perform backup and restore 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 carry out the work of one backup or recovery task.

Because the control file, SPFILE, and datafiles are accessible by any instance, the backup operation of these files is distributed across all the allocated channels. For backups of archived log files, 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 log files to a local directory. When multiple channels are allocated that have access to the archived logs, for each archived log file, RMAN determines which channels have access to that archived log. Then, RMAN groups together the archived logs that can be accessed by a channel and schedules a backup job using that channel.

If each node in the cluster writes the archived log files to ASM, a clustered file system, or other type of shared storage, then each instance has access to all the archived log files. In this case, the backup of the archived log files is distributed across all the allocated channels.

Backing Up Archived Logs

Whether only one node or all nodes perform archived log backups, ensure that all archived logs 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 logs.

You can configure RMAN to automatically delete the 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 prevent an unnecessary outage that might occur if you run out of available disk space.

To configure RMAN to automatically delete the redo log files from disk after they have been safely backed up, when creating or scheduling your database backups:

1. Select **Also back up all archived logs** on disk if you are performing an online backup. There is no need to back up archived logs 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.
2. Select **Delete all archived logs from disk after they are successfully backed up** if you are using shared storage for your archived log files.

> **Note:** Do not select **Delete all archived logs from disk after they are successfully backed up** if you are using a flash recovery area as your *only* archive log destination. In this case, redo log files that have been backed up are deleted automatically as space is needed for storage of other files.

Preparing to Restore and Recover Your Oracle RAC 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 proactively detecting situations such as corrupted database files. Enterprise Managers takes you through the recovery process, prompting for information and performing required recovery actions.

The node that performs the recovery of an Oracle RAC database must be able to restore all the required datafiles. That node must also be able to either read all the required archived log files on disk or be able to restore the archived log files from backup files.

Configuring Access to the Redo Log Files

During recovery, as long as the archived log destinations are visible from the node that performs the recovery, Oracle RAC can successfully recover the archived log files.

If you do not use shared storage or a clustered file system to store the archived log files for your cluster database, then you need to make the archived log files available to the node performing the recovery.

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 crash. Any database file can be vulnerable to a media failure. If a media failure occurs, then you must use 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 will be performing the recovery should be started in `MOUNT` mode. The other instances should be started in `NOMOUNT` mode.

Recovering Your Oracle RAC Database

This section discusses both instance recovery and media recovery.

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 databases, except that you access RMAN from the Maintenance 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. Go to the Cluster Database Home Page and click the **Maintenance** tab.
2. On the Cluster Database Maintenance page, under the Backup/Recovery column, select **Perform Recovery**.
3. Follow the recovery procedures outlined in Chapter 9 of *Oracle Database 2 Day DBA*

Recovering the Parameter File from an Automatic File Backup

You can use Enterprise Manager to restore a lost or damaged server parameter file (SPFILE).

To restore an SPFILE for an Oracle RAC database:

1. With the database in the `MOUNT` state, go to the Backup/Recovery section on the Maintenance tab.
2. Click **Perform Recovery**.

   When the database is not open, the Perform Recovery link takes you to the SPFILE restore page.
3. Restore the SPFILE to either its default location or to a new location that you specify.

> **See Also:** *Oracle Database Backup and Recovery Basics* for more information about restoring a server parameter file

About Restoring Archived Log 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 log file with the sequence number 1001 is backed up to the drive attached to the node `docrac1`, while the archived log file with sequence number 1002 is backed up to the drive attached to the node `docrac2`. If you allocate channels that connect to nodes `docrac1` and `docrac2` for a restore operation, then the channel connected to `docrac1` restores log sequence 1001, but not log sequence 1002. The channel connected to `docrac2` can restore log sequence 1002, but not log sequence 1001.

If you use ASM or a clustered file system for storing the archived redo logs, then any instance can restore the archived redo logs.

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 Datafiles—When restoring datafiles, 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, you can have up to five parallel streams restoring datafiles.
- Applying Incremental Backups—Similarly, when you are applying incremental backups, the number of channels you allocate determines the potential parallelism.
- Applying Archived Redo Logs—With RMAN, the application of archived redo logs is performed in parallel. Oracle RAC automatically selects the optimum degree of parallelism based on available CPU resources.

Managing Your Database Backups

Managing RMAN backups, with or without Enterprise Manager, consists of two tasks: managing the backups of your database that are stored on disk or tape, and managing the record of those backups in the RMAN repository. Enterprise Manager simplifies both backup management tasks.

Some of the tasks involved in managing backups include the following:

- Searching for backups
- Validating the contents of backup sets or image copies
- Cross-checking a backup
- Deleting expired or obsolete backups
- Marking backups as available or unavailable

> **See Also:** *Oracle Database 2 Day DBA*. for more information about these topics and details on how to perform these tasks

Displaying Backup Reports for Your Oracle RAC Database

Backup reports contain summary and detailed information about past backup jobs run by RMAN, including both backups run through Enterprise Manager and the RMAN command-line client.

To view backup reports:

1. From the Cluster Database Home page, click the **Maintenance** tab.
2. On the Maintenance property page, under the Backup/Recovery column, select **Backup Reports**.

   The Backup Reports page contains a list of recent backup jobs.
3. Specify any filter conditions and click **Go** to restrict the list to backups of interest.

   You can use the **Filter By** 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 to be listed (whether it succeeded or failed, and whether warnings were generated during the job).
4. To view detailed information about any backup, click the value in the **Backup Name** column.

   The View Backup Report page is displayed for the selected backup. This page contains summary information about this backup, such as how many files of each type were backed up, how much data total, and the number, and the size and type of backup files created.

   The View Backup Report page also contains a Filter By section that you can use to quickly run a search for another backup or backups from a specific date range. The resulting report contains aggregate information for backups matching the search criteria.

> **See Also:** *Oracle Database 2 Day DBA* for more information about displaying backup reports using Enterprise Manager

Performing Backups and Recovering Your Database: Oracle By Example Series

Oracle By Example (OBE) has a series of tutorials for Oracle RAC databases. This OBE steps you through the basic administrative tasks described in this chapter and includes annotated screen shots.

To view the Performing Backups and Recovering Your Database OBE, go to the following URL

http://www.oracle.com/technology/obe/10gr2_db_vmware/ha/rman/rman.htm

7

Managing Database Workload Using Services

Using workload management, you can distribute the workload across database instances to achieve optimal database and cluster performance for users and applications. This chapter contains the following sections:

- About Workload Management
- Creating Services
- Administering Services
- Transparent Application Failover: Oracle By Example Series

About Workload Management

To implement workload management for an Oracle Real Application Clusters (Oracle RAC) database, you can several different features. This section contains the following topics:

- About Oracle Services
- About the Database Resource Manager
- About Oracle RAC High Availability Framework
- Enabling the Load Balancing Advisory
- About Connection Load Balancing
- About Runtime Connection Load Balancing
- About Fast Application Notification (FAN)

You can deploy Oracle RAC and single-instance Oracle database environments to use workload management 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 several considerations that this chapter describes.

About Oracle Services

Oracle Database 10*g* introduces an automatic workload management facility, called services. A service represents the workload of applications with common attributes, performance thresholds, and priorities. A single service can represent an application, multiple applications or a subset of a single application. A single service can be associated with one or more instances of an Oracle RAC database, and a single

instance can support multiple services. Services provide a single system image to manage competing applications, and they allow each workload to be managed as a single unit.

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.

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 many installations, this may be all you need. For more flexibility in the management of the workload using the database, Oracle Database enables you to create multiple services and specify which database instances offer the services.

Services are integrated with the Resource Manager, which enables you to restrict the resources that are used by a service within an instance. In addition, Oracle Scheduler jobs can run under a service, as opposed to a specific instance.

Configuring Services for High Availability

When you create a service, you define which instances typically support that service. These are known as the *preferred* instances for that service. You can also define other instances to support a service if the service's preferred instance fails. These are known as *available* instances for a service.

When you specify a preferred instance for a service, the service runs on that instance during standard operation. Oracle Clusterware attempts to ensure that the service always runs on all the preferred instances that have been configured for a service. If the instance fails, the service is randomly relocated to one of the available instances. You can also manually relocate the service to an available instance. If you do not specify preferred or available instances when you create a service, then by default every instance in the Oracle RAC database is a preferred instance for that service.

If a service fails over to an available instance, 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 7-7. An example callout script for relocating services back to their preferred instances is available in the Oracle Real Application Clusters Sample Code section on Oracle Technology Network at `http://www.oracle.com/technology/sample_code/products/rac/index.html`

You do not have to specify available instances for a service. However, 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.

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.

About the Database Resource Manager

The Database Resource Manager controls 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 you to map a consumer group to a service so that users who connect

with the service are members of the specified consumer group. The Database Resource Manager allows a single Oracle database running on one or more computers to support multiple applications and mixed workloads with optimal efficiency.

The Database Resource Manager provides the ability to prioritize work within the 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, could take longer to run. This allows for 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 (resource consumer groups).
- *Resource consumer groups* allow the administrator to group user sessions together 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 allocation methods are used by resource consumer groups and resource plans. The database provides the resource allocation methods that are available, but you determine 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 you can create within a resource plan, allow 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.

> **See Also:** *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 Database to maintain components in a running state at all times. If a component fails, 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.

Enabling 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. The Load Balancing Advisory makes recommendations to applications about which instance to direct application requests for a database service to obtain the best performance based on the policy that you have defined for that service.

When the Load Balancing Advisory is enabled, it will account for the following when balancing connection requests across instances:

- Server power differences
- Sessions that are blocked or waiting
- Failures that slow down processing

Using the Load Balancing Advisory prevents sending work to Oracle RAC instances that are overworked, not responding, or not available.

The Load Balancing Advisory informs the application or client about the current performance level that an Oracle RAC database instance is providing for a service. The Load Balancing Advisory also recommends how much of the workload should be sent to that instance.

The best way to take advantage of the Load Balancing Advisory is to use an integrated Oracle client, one that has the Runtime Connection Load Balancing feature. Integrated Oracle clients subscribe to the Load Balancing Advisory FAN events.

You configure your Oracle RAC environment to use the Load Balancing Advisory by defining service-level goals for each service used. This 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 Runtime 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 in the service as well as available bandwidth to the service.
- Throughput—The Load Balancing Advisory measures the percentage of the total response time that the CPU consumes for the service. This goal measures the efficiency of an instance, rather than the response time.

If you do not select the Enable Load Balancing Advisory option, the service-level goal is set to None, which disables load balancing for the service.

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`.

The client uses a connect descriptor to specify the database to which to connect. This connect descriptor contains a protocol and a database service name. When a client requests a connection, the Listener on the server receives the request and forwards the connection to the Oracle database. You can define your connect descriptors in the `tnsnames.ora` file on the client machine, or include them as part of the connection request.

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 of 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:** *Oracle Database 2 Day DBA* for more information about Oracle Net configuration

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 with DBCA, it creates a sample client-side load balancing connection definition in the `tnsnames.ora` file on the server.

Client-side load balancing includes connection failover. With connection failover, if an error is returned from the chosen address, Oracle Net Services will try the next address in the address list until either a successful connection is made or it has exhausted all the addresses in the list.

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 that you want the Listener to use 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 elapsed time under the service. Use the Short connection load balancing goal for applications that have connections of small duration.
- Long—Connections are distributed across instances based on the number of sessions per 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.

Any services created by using DBCA use the Long connection load balancing goal by default.

> **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.

About Runtime Connection Load Balancing

Runtime 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 service level provided by the database instances as indicated by the Load Balacing Advisory FAN events. This provides load balancing at the transaction level instead of load balancing at the time of the initial database connection.

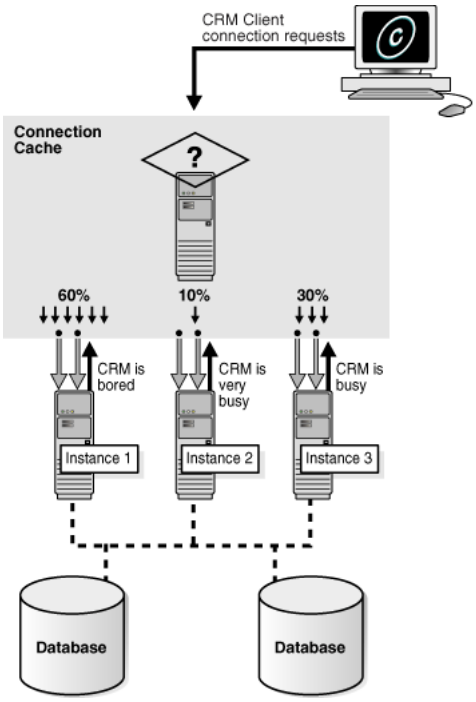
With Runtime Connection Load Balancing, applications use Load Balancing Advisory information to provide better performance to users. The Oracle JDBC and Oracle Data Provider for .NET (ODP.NET) client connection pools are integrated to take advantage of Load Balancing Advisory information. You must enable the client data source for Runtime 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.

Figure 7–1, "Runtime Connection Load Balancing" illustrates Runtime Connection Load Balancing. In this example, 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 Runtime Connection Load Balancing is enabled on the implicit connection cache, the following process occurs:

1. A client requests a connection from the connection cache.
2. Runtime Connection Load Balancing selects the connection that belongs to the best instance from the connection cache. In Figure 7–1, there are three possible nodes the connection can be routed to. Instance1, which has the least amount of CPU workload is currently being assigned about 60% of the incoming connections. Instance2, which is currently overloaded, is only being assigned around 10% of the incoming connections. Instance3, which has a high workload, is being assigned around 30% 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 7–1 Runtime Connection Load Balancing



About Fast Application Notification (FAN)

Fast Application Notification is a 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.

For cluster configuration changes, the Oracle RAC high availability framework publishes a FAN event immediately when a state change occurs 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 to the failed instance or node can be terminated. Incomplete transactions can be terminated and the application user is immediately notified. 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.

About FAN Callouts

FAN callouts are server-side executable files that Oracle RAC executes immediately when high availability events occur. 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
- Executing shell scripts

The executable files for FAN callouts are stored in the `racg/usrco` subdirectory of the Oracle Clusterware home directory. If this subdirectory does not exist in your Oracle Clusterware home, then you must create this directory with the same permissions and ownership as the Oracle Clusterware home `racg/tmp` subdirectory.

A copy of the executable files used by FAN callouts should be available on every node that runs Oracle Clusterware. Example callout scripts are available in the Oracle Real Application Clusters Sample Code section on Oracle Technology Network at http://www.oracle.com/technology/sample_code/products/rac/index.html

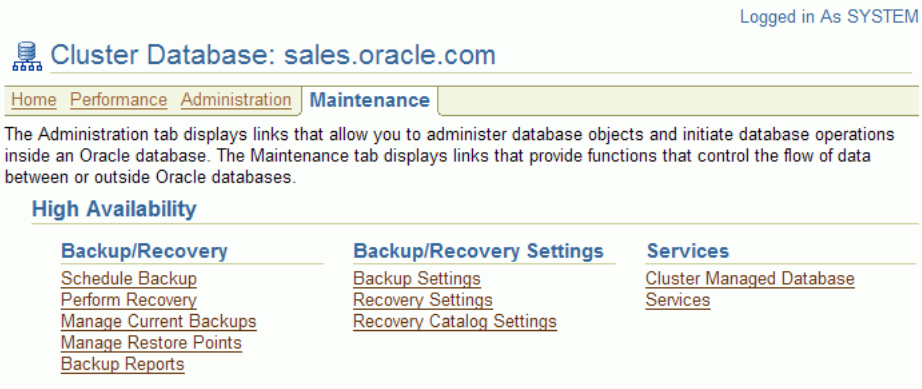
> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring Fast Application Notification and FAN callouts

Creating Services

You can create a service using Oracle Enterprise Manager Database Control.

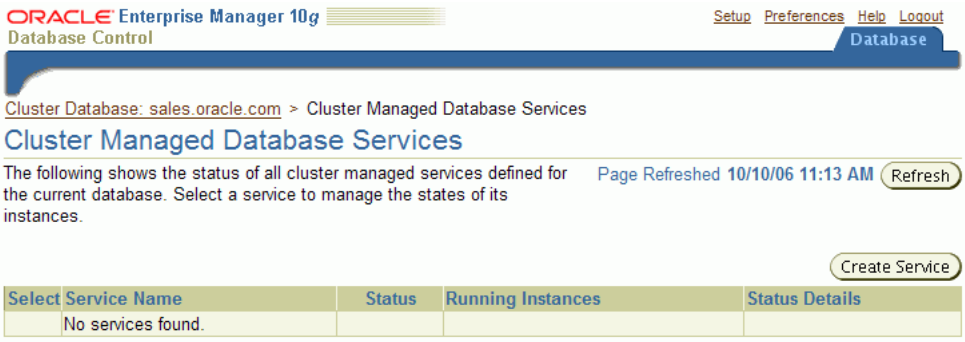
To create a service:

1. On the Cluster Database Home page, click **Maintenance**.



2. Click **Cluster Managed Database Services** in the High Availability section. Enter or confirm the preferred credentials for the Oracle RAC database and host.

   The Cluster Managed Database Services page appears.

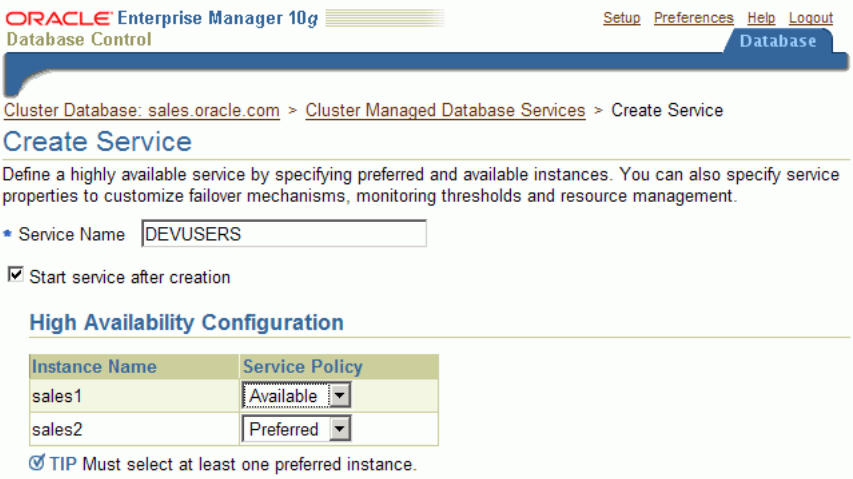


3. Click **Create Service**.

   The Create Service page appears.

4. Enter the name of your service in the Service Name field, for example, `DEVUSERS`.

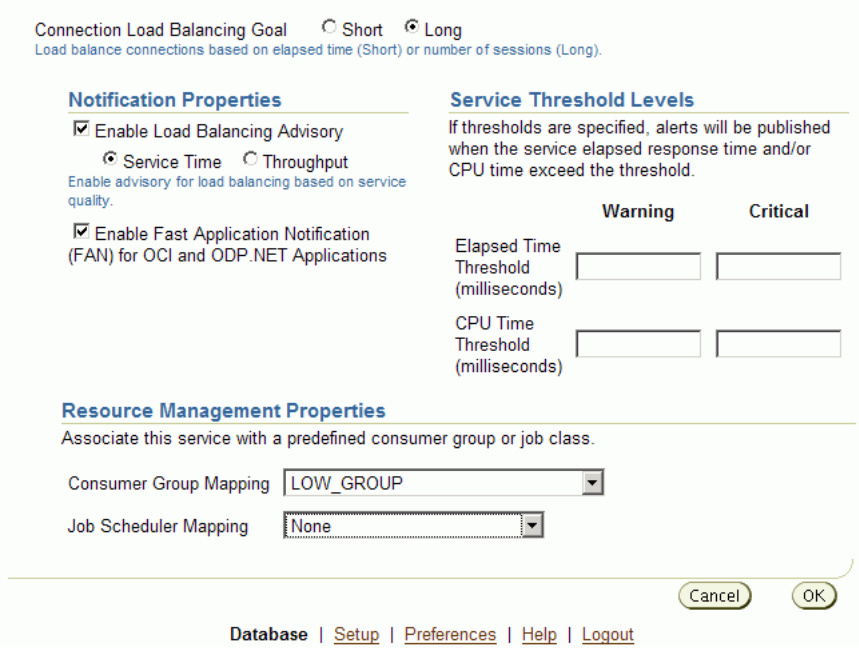
5. If you want the service to be started after it is created, then select **Start Service after creation**.



6. For each instance displayed in the High Availability section, you can choose whether the instance is a Preferred or Available instance for this service. If you do not want the service to run on a particular instance, select **Not Used** for the Service Policy.

7. If you want to modify the connection load balancing goal to distribute the connection workload based on elapsed time instead of the overall number of connections, select **Short**. Otherwise, leave this setting at the default value of Long.

8. To enable the Load Balancing Advisory for this service, select **Enable Load Balancing Advisory** under the heading Notification Properties on the Create Service page, as shown in the following screenshot. Choose a service-level goal of either Service Time or Throughput.



9. If this service is used by an Oracle Call Interface (OCI) or ODP.NET application, and you want to enable FAN, select **Enable Fast Application Notification** under the heading Notification Properties.
10. In the Service Level Thresholds section you can optionally set the service-level thresholds by entering a value in microseconds for Warning and Critical thresholds for the Elapsed Time and CPU Time metrics.
11. If you want to use a Resource Plan to control the resources used by this service, 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.

12. If this service is used by a specific Oracle Scheduler job class, you can specify the mapping by selecting the name from the Job Scheduler Mapping list in the Resource Management Properties section.
13. Click **OK** to create the service.

    See Also: *Oracle Database Administrator's Guide* for more information about the Oracle Scheduler and job classes

Configuring Oracle Net to Support Services

Although Enterprise Manager configures Oracle Clusterware resources for the newly created service, it does not generate the corresponding entries in your `tnsnames.ora` file.

To configure Oracle Net Services to support the newly created service:

1. Determine if the Listener on the local node recognizes the new service by using the following command:

   ```
   lsnrctl status
   ```

   You should see a list for the new service, similar to the following:

   ```
   Service "DEVUSERS.oracle.com" has 1 instance(s).
     Instance "sales1", status READY, has 2 handler(s) for this service...
   ```

   The displayed name for your newly created service, for example `DEVUSERS.oracle.com`, is the value you will use for the `SERVICE_NAME` parameter in the `tnsnames.ora` file.

2. Use a text editor to modify the `tnsnames.ora` file in the `$ORACLE_HOME/network/admin` directory on each node that contains an instance listed as a Preferred or Available instance for the service. Add an entry similar to the following, specifying for the VIP address for each node:

   ```
   DEVUSERS =
     (DESCRIPTION =
       (ADDRESS_LIST = Service
         (ADDRESS = (PROTOCOL = TCP)(HOST = docrac1-vip)(PORT = 1521))
         (ADDRESS = (PROTOCOL = TCP)(HOST = docrac2-vip)(PORT = 1521))
       (LOAD_BALANCE = yes)
       )
     (CONNECT_DATA = (SERVICE_NAME = DEVUSERS.oracle.com))
      )
   ```

   In the previous example, the `ADDRESS_LIST` parameter contains one `ADDRESS` for each node that contains an instance configured as either Preferred or Available for the service.

3. Test the Oracle Net Services configuration by attempting to connect to the database using SQL*Plus and the service name, for example:

   ```
   $ sqlplus system/oracle@DEVUSERS
   ```

   You should see a message indicating you successfully connected to the Oracle database. If you get an error message, examine the `tnsnames.ora` file and verify the entry was typed in correctly and all the information is correct for your environment.

4. Repeat these steps on the other nodes in your cluster that contain instances specified as either Preferred or Available for the newly created service.

Administering Services

You can create and administer services with Enterprise Manager or DBCA. You can also use the `DBMS_SERVICE` PL/SQL package and the SRVCTL utility to perform most service administration tasks.

The following sections describe how to manage services for your cluster database:

- Administering Services with Enterprise Manager
- Configuring Service-Level Thresholds
- Managing Services Using the Database Configuration Assistant

Administering Services with 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

Using the Cluster Managed Database Services Page

When managing services using Enterprise Manager, you use the Cluster Managed Database Services page.

To access the Cluster Managed Database Services page:

1. From the Cluster Database Home page, click the **Maintenance** tab.
2. From the Cluster Database Maintenance page, under the Services heading in the High Availability options list, click **Cluster Managed Database Services**.

   The Cluster and Database Login page appears.
3. Enter credentials for the database and for the cluster that hosts the cluster database, then click **Continue**.

   The Cluster Managed Database Services page appears and displays services that are available on the cluster database instances.

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, and the status for each service.
- Start or stop a service, or enable or disable a service.
- Access the Create Service and Edit Service pages.
- Access the Services Detail page to perform instance-level tasks for a service.
- Test the connection for a service.

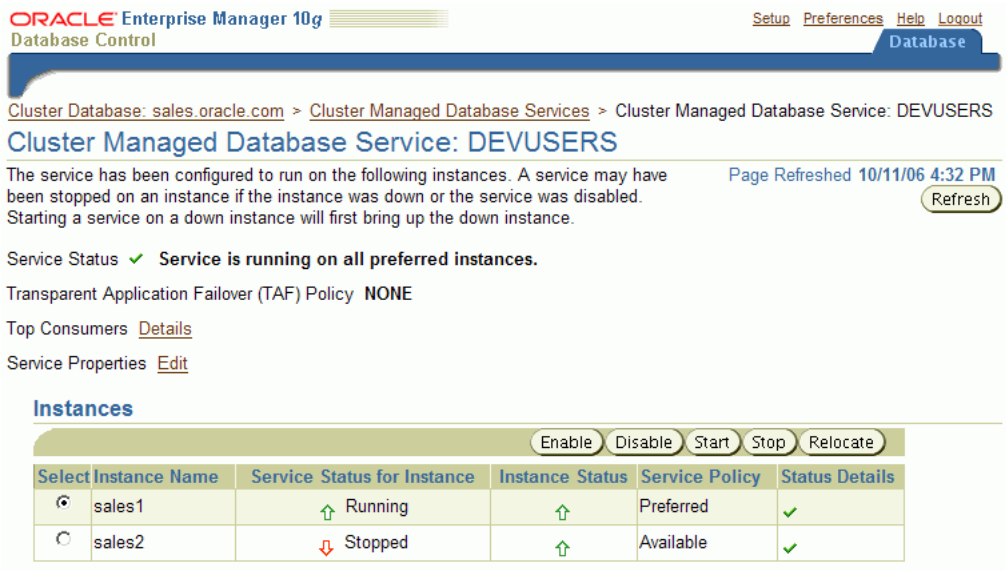
Using the Cluster Managed Database Services Detail Page

You access the Cluster Managed Database Services detail page from the Cluster Managed Database Services page by clicking the name of the service you want to view.

On the Cluster Managed Database Services detail page for a service you can perform the following tasks:

- View the status of a service on all of its preferred and available instances; the status can be Running, Stopped, or Disabled.

- Stop or start a service for an instance of a cluster database.
- Disable or enable a service for an instance of a cluster database.
- Relocate a service to manually rebalance the services across database instances.



Configuring Service-Level Thresholds

When you create a service, you can specify thresholds for measuring the performance of the service. If the specified threshold value is exceeded by the service, the Automatic Workload Repository (AWR) raises an alert that is displayed by Enterprise Manager.

Performance-related statistics, wait events, and active sessions are monitored at the service level. Also, the AWR enables you to monitor performance using services. It records the service performance, including SQL execution times, wait classes, and resources consumed by a service.

You can specify values for the Elapsed Time Threshold or the CPU Time Threshold when you create a service. You can specify Warning and Critical threshold values for these metrics.

To modify service-level thresholds:

1. From the Cluster Database Home page, scroll down to the Instances section.
2. Click the name of the instance for which you want to modify the metric thresholds.
3. On the Cluster Database Instance Home page, in the Related Links section at the bottom of the page, click **Manage Metrics**.
4. Click **Edit Thresholds**.
5. Select either the **Service Response Time** or the **Service CPU Time** threshold.
6. Enter a threshold value in microseconds in the Warning Threshold or Critical Threshold columns, then click **OK**.

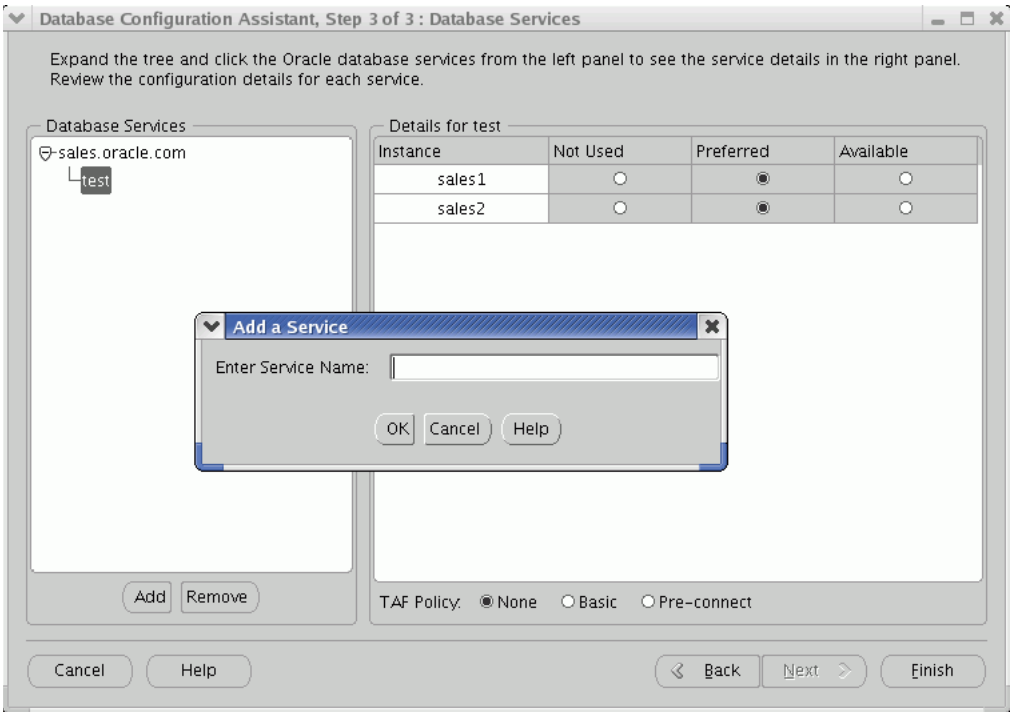
Managing Services Using the Database Configuration Assistant

The Oracle Database Configuration Assistant (DBCA) Services Management feature enables you to manage service assignments and service preferences for instances. You can perform these procedures while your Oracle RAC database is running. Even if your instance or the Oracle RAC database is not running, you can still use DBCA to configure services, but the services will not start automatically.

To add, modify, or delete services using the DBCA Services Management feature:

1. Start the DBCA utility. On Red Hat Linux, the executable file is located in the `$ORACLE_HOME/bin` directory.
2. On the DBCA Welcome window, select the **Oracle Real Application Clusters database** option and click **Next**.
3. On the DBCA Operations window, select **Services Management** and click **Next**.
4. If you have not defined any services, the Database Services window appears. In this window, expand the entry for the cluster database for which you want to manage services.

   If have previously created services for at least one database, the List of cluster databases window appears. Select the name of the database for which you want to configure service, then click **Next**. The Database Services window appears.
5. Add, remove or modify the services.
   - To add a service, click **Add**. In the Add a Service window, enter the name of the service, then click **OK**. Note that service names with the prefix SYS$ are reserved for use by Oracle internal processes.
   - To modify a service, select the available options to configure the service's instance preferences. Assign the service to instances for preferred (normal) and available (recovery) processing. DBCA records your changes when you select another service or proceed to another window. You can also modify the TAF policy for the service.
   - To delete a service, select the service and click **Remove**.



6. Click **Finish**, then click **OK** to confirm you want to configure the services for the database. DBCA displays a progress indicator while it configures your services. When it has completed, you are asked if you want to perform another operation. Select **Yes** to return to the Operations page or **No** to exit DBCA.

When you click Finish, DBCA configures the Oracle Clusterware resources for the services that you added, modified, or removed. DBCA also configures the Oracle Net Services entries for these services and starts them. When you use DBCA to remove services, DBCA stops the service, removes the Oracle Clusterware resource for the service, and removes the Oracle Net Services entries.

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about managing services by using the PL/SQL `DBMS_SERVICE` package procedures or by using the SRVCTL utility

Transparent Application Failover: Oracle By Example Series

When Oracle Net Services establishes a connection to an instance, the connection remains open until the client closes the connection, the instance is shut down, or a failure occurs. If you configure transparent application failover (TAF) for the connection, then Oracle RAC moves the session to a surviving instance when an outage occurs.

TAF can restart a query after failover has completed, but for other types of transactions, such as `INSERT`, `UPDATE`, or `DELETE`, the application must rollback the failed transaction and resubmit the transaction. You must reexecute any session customizations, in other words, `ALTER SESSION` statements, after failover has occurred.

Oracle By Example (OBE) has a series on the High Availability features of Oracle Database 10*g* Release 2. This OBE shows you how to use Enterprise Manager and PL/SQL to set up TAF in an Oracle RAC environment.

To view the Transparent Application Failover OBE tutorial, go to the following URL:

http://www.oracle.com/technology/obe/10gr2_db_vmware/ha/rac/rac.htm

8

Monitoring 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. This chapter focuses 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*.

This chapter includes the following sections:

- Using Performance Views in Oracle Real Application Clusters
- Monitoring Oracle RAC Database Performance
- Using the Automatic Workload Repository in Oracle RAC Environments
- Troubleshooting Configuration Problems in Oracle RAC Environments

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about diagnosing problems for Oracle Clusterware and Oracle Real Application Clusters components

Using Performance Views in Oracle Real Application Clusters

Oracle RAC contains a set of underlying views that are maintained by the database and accessible to the database administrator user `SYS`. These views are called dynamic performance views because they are continuously updated while a database is running and in use, and their contents relate primarily to performance.

Although these views appear to be standard database tables, they are not. These views provide data on internal disk structures and memory structures. You can select from these views, but you can never update or alter them.

GV$ Views

Each instance has a set of instance-specific views, which are prefixed with `V$`. You can also query global dynamic performance views to retrieve performance information from all the qualified instances. Global dynamic performance view names are prefixed with `GV$`.

Querying a `GV$` view retrieves the `V$` view information from all qualified instances. In addition to the `V$` information, each `GV$` view contains an extra column named `INST_ID` of data type `NUMBER`. The `INST_ID` column displays the instance number from which the associated `V$` view information was obtained.

The INST_ID column can be used as a filter to retrieve V$ information from a subset of available instances. For example, the following query retrieves the information from the V$LOCK view for instances 2 and 5:

```
SQL> SELECT * FROM GV$LOCK WHERE INST_ID = 2 OR INST_ID = 5;
```

Monitoring Oracle RAC Database Performance

Both Oracle Enterprise Manager Database Control and Oracle Enterprise Manager Grid Control are cluster-aware and provide a central console to manage your cluster database. From the Cluster Database page, you can do all of the following:

- View the overall system status, such as the number of nodes in the cluster database 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.
- Monitor performance metrics aggregated across all the instances or displayed side by side so you can readily compare 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 through the cache on any instance will return the most recent data written to that location through any cache on any instance.

Enterprise Manager accumulates data over specified periods of time, called collection-based data. Enterprise Manager also provides current data, known as real-time data. The following sections explain how to monitor both types of data:

- Using the Cluster Database Performance Page
- Reviewing the Chart for Cluster Host Load Average
- Reviewing the Chart for Global Cache Block Access Latency
- Reviewing the Cluster Cache Coherency Page
- Reviewing the Chart for Average Active Sessions
- Reviewing the Charts for Database Throughput
- Reviewing Other Performance-Related Charts

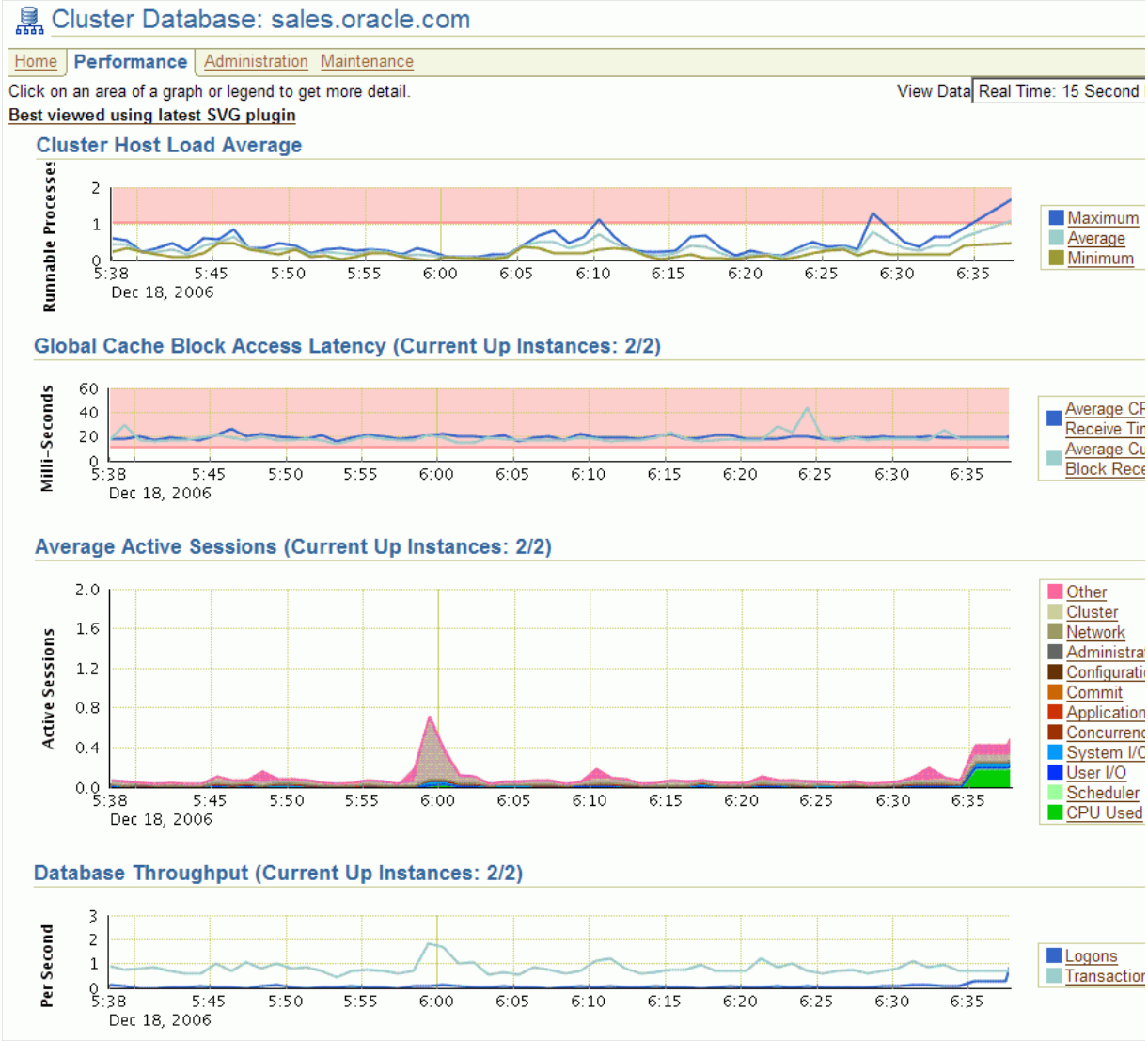
Using the Cluster Database Performance Page

The Cluster Database Performance page provides a quick glimpse of the performance statistics for a database. Statistics are rolled up across all the instances in the cluster database. Using links that are next to the charts at the bottom of the page, you can get more specific information, allowing you to 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 clicking the **Performance** tab from the Cluster Database Home page.



Reviewing the Chart for Cluster Host Load Average

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 hosts 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.

Reviewing the Chart for Global Cache Block Access Latency

Each cluster database instance has its own buffer cache in their System Global Areas (SGAs). 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, 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, 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 `sales1` instance loads a data block into its buffer cache, the `sales2` instance can more quickly acquire the data block from the `sales1` 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, 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, and 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, or the elapsed time it takes to locate and transfer each consistent read and current blocks between the buffer caches.

If the Global Cache Block Access Latency chart shows high latencies, this can be caused by any of the following:

- A high number of requests caused by SQL statements that are not tuned.
- A large number of processes in the queue waiting for 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.

Reviewing the Cluster Cache Coherency Page

To access the Cluster Cache Coherency page, click **Cluster Cache Coherency** in the Additional Monitoring Links section of the Cluster Database Performance page. You can alternatively click either of the legends to the right of the Global Cache Block Access Latency chart.

The Cluster Cache Coherency page appears, as shown in the following screenshot. This page contains summary charts for cache coherency metrics for the cluster.

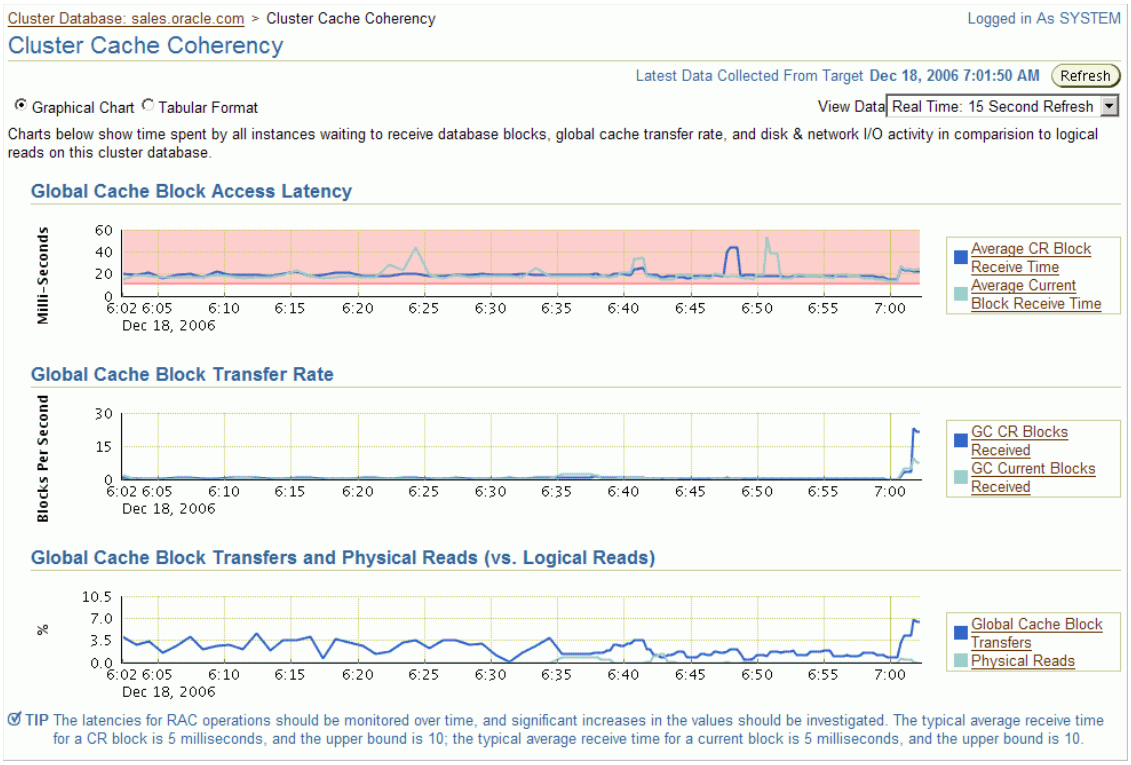


Table 8–1 provides a description of the Cluster Cache Coherency charts and the actions to perform to access more comprehensive information for problem resolution.

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 one of the legends 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 per block class were received from remote instances, including how many transfers incurred a delay (busy) or an unexpected longer delay (congested).

Table 8–1 (Cont.) Cluster Cache Coherency Charts

Name	Description
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 one of the legends 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 that incur some latency overhead. Click the legends 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.

Reviewing the Chart for Average Active Sessions

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 waiting for 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 load on the instance and identifies performance issues. At the cluster database level, this chart shows the aggregate wait class statistics across all the instances.

Compare the peaks on the Average Active Sessions chart with those on the Database Throughput charts. If the Average Active Sessions chart displays a large number of 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, you can view instance-level information stored in Active Sessions by Instance pages. These pages show the service times for up to four instances. Using these pages, if you need to diagnose and fix problems that are causing the higher number of wait events in a specific category, you can select an instance of interest and view the wait events, as well as the SQL, sessions, services, modules, and actions that are consuming the most database resources.

> **See Also:** *Oracle Database 2 Day DBA* for more information about tuning a database and instance

Reviewing the Charts for Database Throughput

The Database Throughput charts summarize any contention that appears in the Average Active Sessions chart, and also show how much work the database is performing on behalf of the user or applications. The Per Second view is for databases that handle SQL queries, shown as Physical Reads in the bottom chart. The Per

Transaction view is for databases that handle transactions, shown as Transactions in the top chart and Redo Size in the bottom chart. Logons show how many users are logged on to the database per second.

You can also obtain information at the instance level by clicking one of the legends to the right of the charts to access the Database Throughput by Instance page. This page shows the breakdown for all active instances of the aggregated Database Throughput chart on the Cluster Database Performance page. You can use this page to view the throughput for a particular instance, which may help you diagnose throughput problems.

You can drill down the list farther 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 page. For more information about this page, refer to the Enterprise Manager Help system.

Reviewing Other Performance-Related Charts

In the Additional Monitoring Links and Additional Instance Monitoring Links sections 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 describes the following charts:

- Top Consumers Page
- Top Sessions Page
- Instance Activity Page
- Top Segments Page
- Database Locks Page

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, clients, and actions that are consuming the most system resources. You access the Top Consumers page by clicking **Top Consumers** in the Additional Monitoring Links section of the Cluster Database Performance page.

By default, the Top Consumers page initially displays the Overview tab, which shows aggregated summary data for the highest resource consumers. For instance-level information about a consumer, click the portion of a chart representing the consumer or click the link under the chart for that consumer. The page that appears shows the running instances that are serving the consumer. You can expand the names in the Module column to show data for individual instances.

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 or not you want to stop the sessions. You access the Top Sessions page by clicking **Top Sessions** in the Additional Monitoring Links section of the Cluster Database Performance page.

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 access top sessions statistics for a particular metric

by clicking a metric legend under the chart if in Graphic mode, or by clicking a name in the summary table if in Tabular mode. You can also use the Switch Database Instance menu to toggle from one database instance to another. You can view data on a per-second or per-transaction basis. You access this page by clicking **Instance Activity** in the Additional Instance Monitoring Links section at the bottom of the Cluster Database Performance page.

Top Segments Page

Collecting and viewing segment-level statistics 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 contention.

You access the Top Segments page by clicking **Top Segments** in the Additional Monitoring Links section. You can view segments for all instances, or use a filter to see segments for a specific instance.

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. You access the Database Locks page by clicking **Database Locks** in the Additional Monitoring Links section.

Using the Automatic Workload Repository in Oracle RAC Environments

In Oracle RAC environments, each Automatic Workload Repository (AWR) snapshot captures data from all active instances within the cluster. The data for each snapshot set that is captured for all active instances is from the same point in time. In addition, the data for each instance is stored separately and is identified with an instance identifier. For example, the `buffer_busy_wait` statistic shows the number of buffer wait events on each instance. AWR does not store data that is aggregated from across the entire cluster. In other words, the data is stored for each individual instance.

AWR automatically generates snapshots of the performance data once every hour and collects the statistics in the workload repository. You can also manually create snapshots, but this is usually not necessary. The data in the snapshot interval is then analyzed by the Automatic Database Diagnostic Monitor (ADDM).

> **See Also:** *Oracle Database Performance Tuning Guide* for more information about Automatic Workload Repository

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 prior to installation. Both Oracle Clusterware and Oracle Database components have subcomponents that you can troubleshoot. The crsctl command `check` enables you to determine the status of several Oracle Clusterware components at one time.

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

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:

- Obtaining Oracle Clusterware Component Names
- Enabling Debugging of Oracle Clusterware Components
- Enabling and Disabling Oracle Clusterware Daemons
- Locating the Oracle Clusterware Alert Log
- Enabling Debugging for an Oracle Clusterware Resource
- Checking the Status of the Oracle Clusterware Installation
- Locating the Oracle Clusterware Component Log Files
- Running the Oracle Clusterware Diagnostics Collection Script

Obtaining Oracle Clusterware Component Names

Use the following command to obtain component names, where *`module_name`* is `crs`, `evm`, `css` or the name of the module:

```
# crsctl lsmodules module_name
```

For example, viewing the components of the `css` module might return the following results:

```
# crsctl lsmodules css
The following are the CSS modules ::
CSSD
COMMCRS
COMMNS
```

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 as follows, where *`component_name`* is the name of an Oracle Clusterware component for which you want to enable debugging, such as `crs`, `evm`, or `css`, *`module`* is the name of module as it appears in the output for the `crcstl lsmodules` command, and *`debugging_level`* is a number from 1 to 5:

```
# crsctl debug log component module:debugging_level
```

For example, to enable tracing for the `CSSD` module of the `css` component, you could use the following command:

```
# crsctl debug log css CSSD:1
```

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 doing this, you can disable them using `crsctl` commands.

Run the following command to enable startup for all the Oracle Clusterware daemons:

```
# crsctl enable crs
```

Run the following command to disable the startup of all the Oracle Clusterware daemons:

```
# crsctl disable crs
```

Locating 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 a CRS resource failed.

The location of the Oracle Clusterware log file is *CRS_home*`/log/`*hostname*`/alert`*hostname*`.log`, where *CRS_home* is the directory in which Oracle Clusterware was installed and *hostname* is the host name of the local node.

Enabling Debugging for an Oracle Clusterware Resource

You can use `crsctl` commands to enable resource debugging using the following syntax, where *resource_name* is the name of an Oracle Clusterware resource, such as `ora.docrac1.vip`, and *debugging_level* is a number from 1 to 5:

```
# crsctl debug log res resource_name:debugging_level
```

To obtain a list of the resources available for debugging, use the following command:

```
# crs_stat
```

Note: When you enable debugging for an Oracle Clusterware resource using `crsctl` commands, this has the same effect as if you set the operating system environment variable `USER_ORA_DEBUG` to 1 before running the start, stop, or check action scripts for the specified resource.

Checking the Status of the Oracle Clusterware Installation

Use the `crsctl check` command to determine the condition of your clusterware installation, as shown in the following example:

```
# crsctl check crs
```

This command displays the status of the Cluster Synchronization Services (CSS), Event Manager (EVM), and the Cluster Ready Services (CRS) daemons. You can also check the status of an individual daemon using the following syntax, where *daemon* is one of `crsd`, `cssd`, or `evmd`:

```
# crsctl check daemon
```

Locating the Oracle Clusterware Component Log Files

Oracle RAC uses a unified log directory structure to store all the Oracle Clusterware component log files. This consolidated structure simplifies diagnostic information collection and assists during data retrieval and problem analysis.

The log files for the CRS daemon, `crsd`, can be found in the following directory:

```
CRS_home/log/hostname/crsd/
```

The log files for the CSS deamon, `cssd`, can be found in the following directory:

```
CRS_home/log/hostname/cssd/
```

The log files for the EVM deamon, `evmd`, can be found in the following directory:

```
CRS_home/log/hostname/evmd/
```

The log files for the Oracle Cluster Registry (OCR) can be found in the following directory:

```
CRS_home/log/hostname/client/
```

The log files for the Oracle RAC high availability component can be found in the following directories:

```
CRS_home/log/hostname/racg/
$ORACLE_HOME/log/hostname/racg
```

Note: Each program that is part of the Oracle RAC high availability component has a subdirectory assigned exclusively for that program. The name of the program subdirectory is the same as the name of the program.

If any of the Oracle Clusterware components generates a core dump file, it is located is a subdirectory of the log directory for that component.

Running the Oracle Clusterware Diagnostics Collection Script

Run the `diagcollection.pl` script as the `root` user to collect diagnostic information from an Oracle Clusterware installation. The diagnostics provide additional information so that Oracle Support Services can resolve problems. Run this script from the operating system prompt as follows, where *CRS_home* is the home directory of your Oracle Clusterware installation:

```
# CRS_home/bin/diagcollection.pl --collect
```

This command displays the status of the Cluster Synchronization Services (CSS), Event Manager (EVM), and the Cluster Ready Services (CRS) daemons.

Using the Cluster Verification Utility to Diagnose Problems

The Cluster Verification Utility (CVU) can assist you in diagnosing a wide variety of configuration problems. Refer to the example of using the CVU in "Installing Oracle Clusterware 10g" on page 3-2.

This section contains the following topics:

- Enabling Tracing

- Checking the Settings for the Interconnect
- Troubleshooting a Node with Status of UNKNOWN
- 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

Enabling Tracing

You can enable tracing by setting the environment variable `SRVM_TRACE` to `true`. After setting this variable to true, run the command that you want to trace. 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. The CVU does not generate trace files unless you enable tracing.

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, the CVU discovers all the available network interfaces if you do not specify an interface on the CVU command line.

To verify the accessibility of the cluster nodes from the local node or from any other cluster node, use the component verification command `nodereach` as follows:

```
cluvfy comp nodereach -n node_list [ -srcnode node ] [-verbose]
```

To verify that the other cluster nodes can be reached from the local node through all the available network interfaces or through specific network interfaces, use the component verification command `nodecon` as follows:

```
cluvfy comp nodecon -n node_list [ -i interface_list ] [-verbose]
```

You can also use the `nodecon` command *without* the `-i` option, as shown in the following example:

```
cluvfy comp nodecon -n all [-verbose]
```

When you issue the `nodecom` command as shown in the previous example, it instructs the CVU to perform the following tasks:

- Discover all the network interfaces that are available on the 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.

You can run the `nodecon` command in verbose mode to identify the mappings between the interfaces, IP addresses, and subnets. To verify the connectivity among the nodes through specific network interfaces, use the `comp nodecon` command with the `-i` option. For example, you can verify the connectivity among the nodes

docrac1, docrac2, and docrac3, through interface eth0 by running the following command:

```
cluvfy comp nodecon -n docrac1, docrac2, docrac3 -i eth0 -verbose
```

Troubleshooting a Node with Status of UNKNOWN

If you run the cluvfy command using the -verbose argument and the CVU responds with UNKNOWN for a particular node, then this is because the CVU cannot determine whether a check passed or failed. The cause of this could be because a node is not reachable, or as a result of any system problem that was occurring on that node at the time that the CVU was performing a check.

The following is a list of possible causes for an UNKNOWN response:

- The node is down.
- Executable files that the CVU requires are missing in the *CRS_home*/bin directory or the $ORACLE_HOME/bin directory.
- The user account that ran the CVU does not have privileges 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 the CVU cannot obtain the operating system resources required to perform its checks.

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Installation Guide for Linux*, or for a different operating system, for more information about using the CVU to troubleshoot your Oracle Clusterware setup

Verifying the Existence of Node Applications

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, use the CVU comp nodeapp command, using the following syntax:

```
cluvfy comp nodeapp [ -n node_list] [-verbose]
```

Verifying the Integrity of Oracle Clusterware Components

To verify the existence of all the Oracle Clusterware components, use the component verification comp crs command, using the following syntax:

```
cluvfy comp crs [ -n node_list] [-verbose]
```

Verifying the Integrity of the Oracle Cluster Registry

To verify the integrity of the Oracle Cluster Registry, use the component verification comp ocr command, using the following syntax:

```
cluvfy comp ocr [ -n node_list] [-verbose]
```

Verifying the Integrity of Your Entire Cluster

To verify that all nodes in the cluster have the same view of the cluster configuration, use the component verification comp clu command, as follows:

```
cluvfy comp clu
```

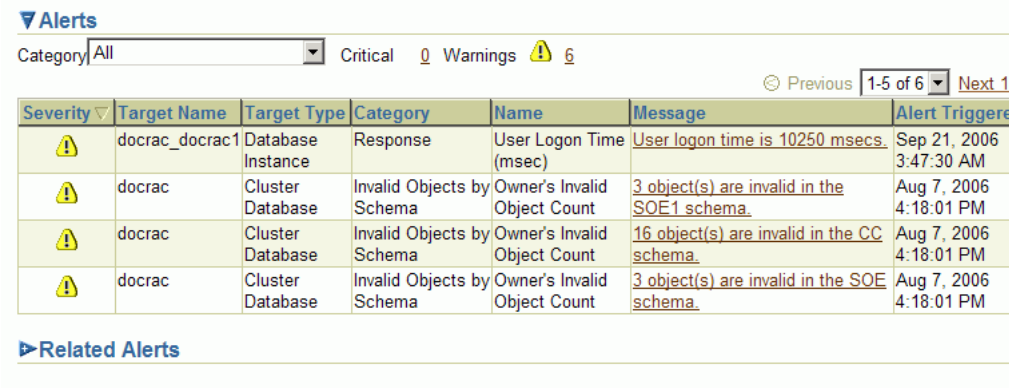
Viewing Oracle RAC Database Alerts

Alert messages are displayed in Enterprise Manager on the Cluster Database Home page under the Alerts heading. The section Related Alerts displays nondatabase alert messages, for example, alert messages for Oracle Net.

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 `sales1` instance exceeded his allotted login time, you would see an alert message with the following values:

- Target name: `sales_sales1`
- Target type: Database instance
- Category: Response
- Name: User logon time
- Message: User logon time is 10250 microseconds
- Alert triggered: *Date and time when the alert condition occurred*

The following screenshot shows an example of the Alerts display for a clustered database named `docrac`.



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

Viewing Oracle RAC Database Alert Log Messages

The alert log is associated with an instance. To view the alert log for your cluster database, you must use the Cluster Database Instance page in Enterprise Manager. On the Home page, in the Diagnostic Summary section, click the date string link next to the heading Alert Log to display the alert log entries containing ORA- errors. To view all the entries in the alert log, click **Alert Log Content** in the Related Links section on the Alert Log Errors page. Enterprise Manager displays the most recent alert log entries by default, but you can specify search criteria to display to alert log entries for a range of dates.

The following screenshot shows an example of the alert log entries for the `docrac1` instance of a cluster database named `docrac`.

ORACLE Enterprise Manager 10g
Database Control

Setup Preferences Help Logout
Database

Database Instance: docrac_docrac1 > Most Recent Alert Log Entries

Search Criteria

Begin Date Time AM PM
(example: Sep 22, 2006)
End Date Time AM PM
(example: Sep 22, 2006)
Go

Most Recent Alert Log Entries

Page Refreshed Sep 22, 2006 2:06:51 PM

This shows the last 100,000 bytes of the alert log. The log is constantly growing, so select the browser's Refresh button to see the most recent log entries.

Number of Lines Displayed **2,187**

```
Thu Aug 10 15:05:50 2006
Thread 1 advanced to log sequence 130
  Current log# 1 seq# 130 mem# 0: +DATA/docrac/onlinelog/group_1.262.594989193
  Current log# 1 seq# 130 mem# 1: +DATA/docrac/onlinelog/group_1.263.594989203
Thu Aug 10 21:02:03 2006
Thread 1 advanced to log sequence 131
  Current log# 2 seq# 131 mem# 0: +DATA/docrac/onlinelog/group_2.264.594989215
  Current log# 2 seq# 131 mem# 1: +DATA/docrac/onlinelog/group_2.265.594989225
```

9

Adding Nodes and Instances

This chapter describes how to add nodes and instances in Oracle Real Application Clusters (Oracle RAC) environments. You can use these methods when configuring a new Oracle RAC cluster, or when scaling up an existing Oracle RAC cluster.

This chapter includes the following sections:

- Preparing Access to the New Node
- Extending the Oracle Clusterware Home Directory
- Extending the Oracle Automatic Storage Management Home Directory
- Extending the Oracle RAC Software Home Directory
- Creating a Listener on the New Node
- Adding a New Cluster Instance on the New Node

For this chapter, it is very important that you perform each step in the order shown.

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about adding and removing nodes from your cluster database

Preparing Access to the New Node

To prepare the new node prior to installing the Oracle software, refer to Chapter 2, "Preparing Your Cluster".

It is critical that you follow the configuration steps for the following procedures to work. These steps include, but are not limited to the following:

- Adding the public and private node names for the new node to the `/etc/hosts` file on the existing nodes, `docrac1` and `docrac2`
- Verifying the new node can be accessed (using the `ping` command) from the existing nodes
- Running the following command on either `docrac1` or `docrac2` to verify the new node has been properly configured:

  ```
  cluvfy stage -pre crsinst -n docrac3
  ```

Extending the Oracle Clusterware Home Directory

Now that the new node has been configured to support Oracle Clusterware, you use Oracle Universal Installer (OUI) to add an Oracle Clusterware home to the node being added to your Oracle RAC cluster. This chapter assumes that you are adding a node named `docrac3` and that you have already successfully installed Oracle Clusterware on `docrac1` in a nonshared home, where *`CRS_home`* represents the successfully installed Oracle Clusterware home.

To extend the Oracle Clusterware installation to include the new node:

1. Verify the `$ORACLE_HOME` environment variable on `docrac1` directs you to the successfully installed Oracle Clusterware home on that node.
2. Go to `CRS_home/oui/bin` and run the `addNode.sh` script.

   ```
   cd /opt/oracle/crs/oui/bin
   ./addNode.sh
   ```

   OUI starts and first displays the Welcome window.
3. Click **Next**.

   The Specify Cluster Nodes to Add to Installation window appears.
4. Select the node or nodes that you want to add. After selecting `docrac3`, click **Next**.
5. Verify the entries that OUI displays on the Summary Page and click **Next**.
6. Run the `rootaddNode.sh` script from the `CRS_home/install/` directory on `docrac1` when prompted to do so.

   Basically, this script adds the node applications of the new node to the OCR configuration.
7. Run the `orainstRoot.sh` script on the node `docrac3` if OUI prompts you to do so.
8. Run the `CRS_home/root.sh` script on the node `docrac3` to start Oracle Clusterware on the new node.
9. Add the new node's Oracle Notification Services (ONS) configuration information to the shared Oracle Cluster Registry (OCR). Obtain the ONS port identifier used by the new node, which you need to know for the next step, by running the following command from the `CRS_home/opmn/conf` directory on the `docrac1` node:

   ```
   cat ons.config
   ```

   After you locate the ONS port number for the new node, you must make sure that the ONS on `docrac1` can communicate with the ONS on the new node, `docrac3`.
10. From the `CRS_home/bin` directory on the node `docrac1`, run the Oracle Notification Services configuration utility as shown in the following example, where *`remote_port`* is the port number from step 9, and `docrac3` is the name of the node that you are adding:

    ```
    ./racgons add_config docrac3:remote_port
    ```

At the end of the cloning process, you should 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 as the `root` user on the newly configured node, `docrac3`:

```
CRS_home/bin/cluvfy stage -post crsinst -n docrac3 -verbose
```

Extending the Oracle Automatic Storage Management Home Directory

To extend an existing Oracle RAC database to a new node, you must configure the shared storage for the new database instances that will be created on new node. You must configure access to the same shared storage that is already used by the existing database instances in the cluster. For example, the `sales` cluster database in this guide uses Oracle Automatic Storage Management (ASM) for the database shared storage, so you must configure ASM on the node being added to the cluster.

Because you installed ASM in its own home directory, you must configure an ASM home on the new node using OUI. The procedure for adding an ASM home to the new node is very similar to the procedure you just completed for extending Oracle Clusterware to the new node.

To extend the ASM installation to include the new node:

1. Ensure that you have successfully installed the ASM software on at least one node in your cluster environment. To use these procedures as shown, your `$ASM_HOME` environment variable must identify your successfully installed ASM home directory.
2. Go to the `$ASM_HOME/oui/bin` directory on `docrac1` and run the `addNode.sh` script.
3. When OUI displays the Node Selection window, select the node to be added (`docrac3`), then click **Next**.
4. Verify the entries that OUI displays on the Summary window, then click **Next**.
5. Run the `root.sh` script on the new node, `docrac3`, from the ASM home directory on that node when OUI prompts you to do so.

You now have a copy of the ASM software on the new node.

Extending the Oracle RAC Software Home Directory

Now that you have extended the Oracle Clusterware and ASM homes to the new node, you must extend the Oracle Database home on `docrac1` to `docrac3`. The following steps assume that you have already completed the previous tasks described in this chapter, and that `docrac3` is already a member node of the cluster to which `docrac1` belongs.

The procedure for adding an Oracle RAC home to the new node is very similar to the procedure you just completed for extending ASM 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, your `$ORACLE_HOME` environment variable must identify your successfully installed Oracle RAC home directory.
2. Go to the `$ORACLE_HOME/oui/bin` directory on `docrac1` and run the `addNode.sh` script.
3. When OUI displays the Specify Cluster Nodes to Add to Installation window, select the node to be added (`docrac3`), then click **Next**.

4. Verify the entries that OUI displays in the Cluster Node Addition Summary window, then click **Next**.
5. Run the `root.sh` script on the new node, `docrac3`, from the `$ORACLE_HOME` directory on that node when OUI prompts you to do so.

After completing these steps, you should have an installed Oracle RAC home on the new node.

Creating a Listener on the New Node

To service database instance connection requests on the new node, you must create a Listener on that node. Use the Oracle Net Configuration Assistant (NETCA) to create a Listener on the new node. Before beginning this procedure, ensure that your existing nodes have the `$ORACLE_HOME` environment variable set correctly.

To create a new Listener on the new node using Oracle Net Configuration Assistant:

1. Start the Oracle Net Configuration Assistant by entering `netca` at the system prompt from the `$ORACLE_HOME/bin` directory.

   NETCA displays the Welcome window. Click **Help** on any NETCA window for additional information.
2. Select **Listener configuration**, and click **Next**.

   NETCA displays the Listener Configuration, Listener window.
3. Select **Add** to create a new Listener, then click **Next**.

   NETCA displays the Listener Configuration, Listener Name window.
4. Accept the default value of `LISTENER` for the Listener name by clicking **Next**.

   NETCA displays the Listener Configuration, Select Protocols window.
5. Choose TCP and move it to the Selected Protocols area, then click **Next**.

   NETCA displays the Listener Configuration, TCP/IP Protocol window.
6. Choose **Use the standard port number of 1521**, then click **Next**.

   NETCA displays the Real Application Clusters window.
7. Select **Cluster configuration** for the type of configuration to perform, then click **Next**.

   NETCA displays the Real Application Clusters, Active Nodes window.
8. Select the name of the node you are adding, for example `docrac3`, then click **Next**.

   NETCA creates a Listener using the configuration information provided. You can now exit NETCA.

You should now have a Listener named LISTENER running on the new node.

At this point, you should perform any needed service configuration procedures for the new database instance as described inChapter 7, "Managing Database Workload Using Services".

> **See Also:** *Oracle Database Net Services Administrator's Guide* for more information about configuring a Listener using Oracle Net Configuration Assistant

Adding a New Cluster Instance on the New Node

You can use the Oracle Database Configuration Assistant (DBCA) to add database instances to new nodes. Before beginning this procedure, ensure that your existing nodes have the `$ORACLE_HOME` environment variable set correctly.

To create a new cluster instance on the new node using DBCA:

1. Start DBCA by entering `dbca` at the system prompt from the `$ORACLE_HOME/bin` directory.

   DBCA displays the Welcome window for Oracle RAC. Click **Help** on any DBCA page for additional information.

2. Select **Oracle Real Application Clusters database**, and then click **Next**.

   DBCA displays the Operations window.

3. Select **Instance Management**, and then click **Next**.

   DBCA displays the Instance Management window.

4. Select **Add an Instance**, then click **Next**.

   DBCA displays the List of Cluster Databases window, which shows the databases and their current status, such as ACTIVE or INACTIVE.

5. In the List of Cluster Databases window, select the active Oracle RAC database to which you want to add an instance, for example `sales`. Enter the user name and password for the database user that has `SYSDBA` privileges. Click **Next**.

   DBCA will spend a few minutes performing tasks in the background, then it will display the Instance naming and node selection window.

6. In the Instance naming and node selection window, enter the instance name in the field at the top of this window if the default instance name provided by DBCA does not match your existing instance naming scheme. For example, instead of the `sales3` instance, you might want to create the `sales_03` instance.

   Click **Next** to accept the default instance name of `sales3`.

   DBCA displays the Instance Storage window.

7. In the Instance Storage window, you have the option of changing the default storage options and file locations for the new database instance. In this example, you accept all the default values and click **Finish**.

   DBCA displays the Summary window.

8. Review the information in the Summary window, then click **OK** to start the database instance addition operation. DBCA displays a progress dialog box showing DBCA performing the instance addition operation.

9. During the instance addition operation, if you are using ASM for your cluster database storage, DBCA detects the need for a new ASM instance on the new node.

   When DBCA displays a dialog box, asking if you want to ASM to be extended, click **Yes**.

   After DBCA extends ASM on the new node and completes the instance addition operation, DBCA displays a dialog box asking whether or not you want to perform another operation. Click **No** to exit DBCA.

You should now have a new cluster database instance and ASM instance running on the new node. After you terminate your DBCA session, you should run the following

command to verify the administrative privileges on the new node and obtain detailed information about these privileges:

```
CRS_home/bin/cluvfy comp admprv -o db_config -d oracle_home -n docrac3 -verbose
```

10

Managing Oracle Software and Applying Patches

Oracle issues product fixes for its software called patches. When you apply the patch to your Oracle software installation, a small collection of files are replaced to fix certain bugs. OPatch is an Oracle supplied utility that facilitates Oracle software patching.

A group of patches form a patch set. When you apply a patch set, many different files and utilities are modified. This results in a version change for your Oracle software, for example, from Oracle Database 10.2.0.1.0 to Oracle Database 10.2.0.2.0. To apply a patch set you use the Oracle Universal Installer (OUI).

This chapter describes how to manage Oracle software and apply patches in Oracle Real Application Clusters (Oracle RAC) environments using Enterprise Manager and the OPatch utility.

This chapter includes the following sections:

- Configuring the Enterprise Manager Patch Interface
- Obtaining the Patch
- Preparing to Use OPatch
- Applying Patches
- Applying Patch Sets
- Troubleshooting Patch Deployment

> **See Also:** *Oracle Universal Installer and OPatch User's Guide* for more information about using OPatch and applying patches to Oracle RAC

Configuring the Enterprise Manager Patch Interface

Oracle Enterprise Manager Database Control enables you to find the latest patch release on the Oracle Meta*Link* Web site and to download it to your Oracle home. There are two steps in configuring the Enterprise Manager Patch interface:

- Setting Oracle MetaLink Credentials
- Running the Refresh_From_Metalink Job

Setting Oracle Meta*Link* Credentials

To download patches from Oracle Meta*Link* using Enterprise Manager, you can give Oracle Enterprise Manager Database Control (Database Control) your login

credentials so that it can log in to Oracle Meta*Link* automatically and search for patch releases. You must set these credentials before you can run the Patch wizard in Database Control.

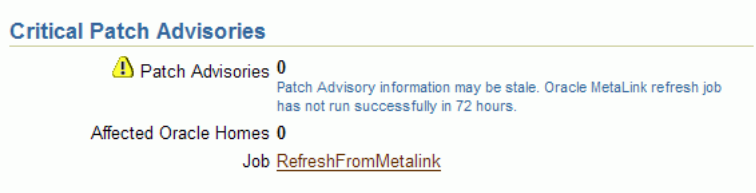
Refer to *Oracle Database 2 Day DBA* for instructions on setting your Oracle Meta*Link* credentials.

Running the Refresh_From_Metalink Job

After you have configured the Oracle Meta*Link* credentials, you can create a job to search for critical patch advisories for your installed software.

To create a job to search for critical patch advisories on Oracle MetaLink, perform the following tasks:

1. On the Cluster Database Home page, scroll down the section titled Critical Patch Advisories. Click the link **RefreshFromMetalink**.



   When you click this link, Enterprise Manager creates the Refresh_From_Metalink_Job job, and then displays the Job Activity page.

2. On the Job Activity page, click **Edit** and then modify the scheduled execution time of the Refresh_From_Metalink_Job job to meet your business requirements. When finished, click **Save**.
3. When returned to the Job Activity page, select the Refresh_From_Metalink_Job job and click **Create Like**. Change the job now to Refresh_From_Metalink_Now, then click **Schedule**. Select **Immediately** for the start time. Select **One Time Only** for the Repeat interval. Click **Submit and Save**.
4. When returned to the Job Activity page, click **REFRESH_FROM_METALINK_NOW**.

   The Job Run: REFRESH_FROM_METALINK_NOW page is displayed.
5. Refresh this page until the job status shows Succeeded.
6. Click the **Database** tab in the upper right-hand corner to return to the Cluster Database Home page.

Obtaining the Patch

You obtain patches and patch sets from Oracle Meta*Link*, which is the Oracle Support Services Web site, at

`http://metalink.oracle.com`

Refer to *Oracle Database 2 Day DBA* for information about configuring Enterprise Manager to access OracleMeta*Link* for patch queries and downloads.

You can view available patch releases at Oracle Meta*Link* by using Enterprise Manager. Viewing these updates is the first step in the Patch wizard, which you can use to download the patch to your Oracle home.

To start the Patch wizard with Enterprise Manager:

1. From the Cluster Database home page, click **Maintenance**.
2. Scroll down to the list of instances. Click the link for the first instance in your cluster, for example, **sales.oracle.com_sales1**.

   The Database Instance Home page for the sales1 instance is displayed.
3. Click **Maintenance**.
4. In the Database Software Patching section, click **Apply Patch**.

   The Patch: Select Patch page appears.
5. Enterprise Manager automatically fills in values for Product Family, Product, Release, and Platform. The search results that match the criteria are displayed by the most recent patch (or patch set) at the top of the list.

   > **Note:** You can limit your search by entering values for Begin Date and End Date, or by specifying a bug number, and then clicking **Search**.

6. Select a patch and click **View Details** to view the patch details. Select a patch and click **View ReadMe** to view the `README` file for the patch, which includes a description of the bug fixes included in the patch and patch installation instructions. Return to the Patch Wizard by clicking the Patch locator link on the View Patch Details page.
7. Select the patch you are interested in, or select the most recent patch set if you are doing a periodic software update, then click **Next**.

   The Patch: Select Destination page appears.
8. Select the targets to apply the patch to by moving the target names from the Available Targets list to the Selected Targets list, then click **Next**.

   The Patch: Set Credentials page appears.
9. In the Username and Password boxes, enter the operating system user name and password to enable Enterprise Manager to stage the patch in your Oracle home directory. Enterprise Manager requires these credentials for job scheduling. After you have entered the operating system credentials for each selected node, click **Next**.

   The Patch: Stage or Apply page appears.
10. Enterprise Manager downloads the patch to the directory that is listed in the main box. Typically, this location is an Oracle home subdirectory called `EMStagedPatches/`*`patchnumber`*.

    By default, Enterprise Manager only stages the patch. You can then manually apply the patch by following the directions given in the patch `README` file. The directions may include shutting down the database instances and your applications, or running scripts.

    You can also select the Run Script to Apply Patch option to have Enterprise Manager apply the patch for you. If you choose this option, you must modify the script displayed on this page so that it performs all the actions specified in the patch release notes. This method is only supported for databases that do not contain the Enterprise Manager repository or for patches that do not require the

repository database to be shutdown. For example, if you are applying a patch that only affects SQL*Loader, then you can use Enterprise Manager to apply the patch.

When you are done, click **Next**. The Patch: Schedule page appears.

11. Specify the time when you want the patch to be downloaded from Oracle Meta*Link*. If you selected the option Run Script to Apply Patch, then the patch apply script will run at this time. Click **Next**.

    The Patch: Summary page appears.

12. Review the summary information on this page. If you need to modify any of the information displayed, click the **Back** button. When you are ready to submit the job, click **Finish**.

Preparing to Use OPatch

Before you apply the patch to your Oracle RAC database, your Oracle ASM installation, or to your Oracle Clusterware installation, there are a few steps to perform:

- Check ORACLE_HOME Environment Variable
- Perform a Backup
- Stage the Patch on Each Node
- Configure SSH User Equivalency

Check 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. Check the respective vendor documentation for the details to set the environment variable.

Perform a Backup

It is highly recommended to back up the software directory you are patching before performing any patch operation. This applies to Oracle Database, Oracle ASM, or Oracle Clusterware software installation directories. You can back up the software installed in the specified `ORACLE_HOME` using any method such as `zip`, `cp -r`, `tar`, and `cpio` to compress the `ORACLE_HOME`.

Stage the Patch on Each Node

If you use Enterprise Manager to download the patch, and you selected all the nodes in your cluster as targets for the patch, then the patch is automatically staged on those nodes. If you manually downloaded the patch from Oracle Meta*Link*, then you must copy the patch to each node.

Update 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. For example, on RedHat Linux systems you would use a shell command similar to the following:

```
$ export PATH=$PATH:/opt/oracle/10gR2/db_1/OPatch
```

You could also modify the shell profile script to have this variable configured every time you log in.

Configure SSH User Equivalency

Before you patch a system, make sure the user equivalency is working. You can use the following command to test user equivalency:

```
[oracle@docrac1] $ ssh docrac2 date
```

If the date is returned, then user equivalency between the source and destination node has been configured. If you see output similar to the following, then SSH user equivalency is not enabled:

```
Enter passphrase for key '/home/oracle/.ssh/id_rsa':
```

To enable SSH User Equivalency:

1. On the system where you want to run OPatch, 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
   ```

   At the prompt, enter the pass phrase for each key that you generated when configuring Secure Shell, for example:

   ```
   [oracle@docrac1 .ssh]$ exec /usr/bin/ssh-agent $SHELL
   [oracle@docrac1 .ssh]$ /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)
   ```

   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. Refer to "Configuring SSH User Equivalency" on page 2-9 for more information about configuring SSH user equivalency.

   To test if you have configured SSH correctly, then you can the following command without being prompted for a password or a pass phrase:

   ```
   [oracle@docrac1] $ ssh docrac2 date
   ```

   Note: Do not close this terminal window until you have completed the patch installation. If you must close this terminal window before the patch installation is complete, repeat step 1 and step 2 before starting the patch installation.

Applying Patches

Patching in an Oracle RAC environment is slightly different compared to patching a single node. If OPatch detects a cluster, it uses Oracle Universal Installer 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 an Oracle RAC 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 Database Control, Listener, and any other applications that are running from the Oracle RAC home directory
Oracle ASM	Oracle RAC database, any single-instance databases that use the same Oracle ASM instance as the cluster database, Listener (if running from the Oracle ASM home directory), Oracle ASM, and any other applications that are running from the Oracle ASM home directory
Oracle Clusterware	Oracle RAC database, any single-instance databases that use the same Oracle ASM instance as the cluster database, Oracle ASM, all node applications, Oracle Clusterware, and any other applications that are running from the Oracle Clusterware home directory

You can patch Oracle RAC in three different ways:

- All Node Patching
- Rolling Patching
- Minimum Downtime Patching

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 all the `nodeapps` on the nodes are restarted. 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.

To implement all node patching:

1. Stop all user applications that use the Oracle RAC home directory.
2. If you are patching only the Oracle RAC home directory, shut down all Oracle RAC instances on all nodes in the cluster. To shut down all Oracle RAC instances for a cluster database, enter the following command where `ORA_CRS_HOME` is the location of the Oracle Clusterware home directory and `sales` is the name of the database:

   ```
   $ $ORA_CRS_HOME/bin/srvctl stop database -d sales
   ```

3. If you are patching the Oracle ASM or Oracle Clusterware home directory, stop all single-instance databases that are running on the group of nodes being patched if they use the Oracle ASM installation that you are patching.
4. If you are patching the Oracle ASM home directory, stop all user applications that use the Oracle ASM home directory on the group of nodes being patched.
5. If you are patching the Oracle ASM or Oracle Clusterware home directory, you can use a single command to stop all the node applications on each node in the

group. This command shuts down the Oracle RAC instances, the Listener, the Oracle ASM instances, and the Oracle Clusterware node applications for the specified node. Use a command similar to the following, where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation and *`node_name`* is the name of the node:

```
$ $ORA_CRS_HOME/crs/bin/srvctl stop nodeapps -n node_name
```

Repeat the above command for each node in the cluster.

After you have stopped the `nodeapps` on each node in the cluster, use the `crs_stat` utility to verify that all the `nodeapps` were stopped on each node.

```
$ $ORA_CRS_HOME/bin/crs_stat -t
```

6. If you are patching the Oracle Clusterware home directory, shut down the CRS daemons for all the nodes in the cluster by issuing the following command as the `root` user on each node, where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation:

   ```
   # $ORA_CRS_HOME/bin/crsctl stop crs
   ```

   Repeat this command on each node.

7. Set your current directory to the directory where the patch is located, for example:

   ```
   $ cd $ORACLE_HOME/EMStagedPatches/4519934/4519934
   ```

8. Make sure the `ORACLE_HOME` environment variable points to the software directory you want to patch, for example:

   ```
   $ echo $ORACLE_HOME
   /opt/oracle/10gR2/db_1
   ```

9. Run OPatch by entering the following command:

   ```
   opatch apply
   ```

10. If you applied the patch to the Oracle Clusterware home directory, restart the CRS daemons on all nodes by issuing the following command as the `root` user on each node, where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation:

    ```
    # $ORA_CRS_HOME/bin/crsctl start crs
    ```

    Repeat this command on each node in the cluster.

11. If you stopped the `nodeapps` on each node, after the patch has been applied, restart the `nodeapps` on all nodes. To start the `nodeapps`, enter a command similar to the following where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation and `docrac1` is one of the nodes in your cluster:

    ```
    $ $ORA_CRS_HOME/bin/srvctl start nodeapps -n docrac1
    ```

    Repeat the above command for each node in the group.

    After you have restarted the `nodeapps` on all nodes, use the `crs_stat` utility to verify that the `nodeapps` were restarted on each node.

    ```
    $ $ORA_CRS_HOME/bin/crs_stat -t
    ```

    If any of the node applications did not restart, use the SRVCTL utility to restart them. For example, you can use commands similar the following to restart various

nodeapps, where ORA_CRS_HOME is the home directory of your Oracle Clusterware installation:

```
$ $ORA_CRS_HOME/bin/srvctl start instance -d sales -i "sales1"
$ $ORA_CRS_HOME/bin/srvctl start listener -n docrac1
$ $ORA_CRS_HOME/bin/srvctl start asm -n docrac1
```

12. Run any post-patch scripts that are mentioned in the patch instructions, for example:

```
$ sqlplus /nolog
SQL> connect sys/password@sales1 AS SYSDBA
SQL> @$ORACLE_HOME/cpu/CPUOct2006/catcpu.sql
SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql
SQL> exit
```

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, Oracle ASM, or Oracle Clusterware installation. By patching groups of nodes individually, there is zero downtime for the cluster database because at least one instance is available at all times on a different node.

While most patches can be applied in a rolling fashion some patches can not be applied in this fashion. The README file for the patch indicates whether or not 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" or "All Node Patching" to apply the patch.

To apply a patch using the rolling patch method:

1. Change to the directory where the unzipped patch is staged on disk, for example:

```
$ cd $ORACLE_HOME/EMStagedPatches/4519934/4519934
```

2. Stop all user applications that use the Oracle RAC home directory for the group of nodes being patched. For example, to stop Enterprise Manager Database Control on the local node, use the following command, where ORACLE_HOME is the home directory for your Oracle RAC installation:

```
$ $ORACLE_HOME/bin/emctl stop dbconsole
```

3. If you are patching the Oracle ASM or Oracle Clusterware home directory, stop all single-instance databases that are running on the group of nodes being patched if they use the Oracle ASM software you are patching.
4. If you are patching the Oracle ASM home directory, stop all user applications that use the Oracle ASM home directory on the group of nodes being patched.
5. If you are patching only the Oracle RAC home directory, shut down all Oracle RAC instances in the group of nodes being patched. To shut down an instance for an Oracle RAC database, enter a command similar to the following example, where ORA_CRS_HOME is the home directory for your Oracle Clusterware installation, sales is the name of the database and sales1 is the name of the instance:

```
$ $ORA_CRS_HOME/bin/srvctl stop instance -d sales -i "sales1"
```

Repeat the above command for each node in the group of nodes being patched.

6. If you are patching the Oracle ASM or Oracle Clusterware home directory, you can use a single command to stop all the node applications on each node in the group. This command shuts down the Oracle RAC instances, the Listener, the Oracle ASM instances, and the Oracle Clusterware node applications for the specified node. Use a command similar to the following, where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation and `docrac1` is one of the nodes in the group:

   ```
   $ $ORA_CRS_HOME/crs/bin/srvctl stop nodeapps -n docrac1
   ```

   Repeat the above command for each node in the group of nodes being patched.

   After you have stopped the `nodeapps` on each node in the group, use the `crs_stat` utility to verify that all the `nodeapps` were stopped on the group of nodes being patched.

   ```
   $ $ORA_CRS_HOME/bin/crs_stat -t
   ```

7. If you are patching the Oracle Clusterware home directory, shut down the CRS daemons for the nodes in the group by issuing the following command as the `root` user on each node in the group, where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation:

   ```
   # $ORA_CRS_HOME/bin/crsctl stop crs
   ```

   Repeat this command on each node in the first group.

8. Make sure the `ORACLE_HOME` environment variable points to the software directory you want to patch, for example:

   ```
   $ echo $ORACLE_HOME
   /opt/oracle/10gR2/db_1
   ```

9. If you are patching nodes individually, use the following command to instruct OPatch to apply the patch to only the local node. If you run this command from the directory where the patch is located, you do not need to specify the patch ID.

   ```
   $ opatch apply -local
   ```

   If you are using a group of nodes, use a command similar to the following to instruct OPatch to apply the patch to the group of nodes being patched:

   ```
   $ opatch apply -local_node docrac1 -remote_nodes docrac2,docrac3
   ```

10. If you applied the patch to the Oracle Clusterware home directory, restart the CRS daemons for the nodes in the first group by issuing the following command as the `root` user on each node in the group, where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation:

    ```
    # $ORA_CRS_HOME/bin/crsctl start crs
    ```

    Repeat this command on each node in the group.

11. If you stopped the `nodeapps` for the group of nodes, after the patch has been applied, restart the `nodeapps` on those nodes. To start the `nodeapps`, enter a command similar to the following where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation and `docrac1` is one of the nodes in the group of nodes you recently patched:

    ```
    $ $ORA_CRS_HOME/bin/srvctl start nodeapps -n docrac1
    ```

Repeat the above command for each node in the group.

After you have restarted the `nodeapps` on each node in the group, use the `crs_stat` utility to verify that the `nodeapps` were restarted on each node in the group.

```
$ $ORA_CRS_HOME/bin/crs_stat -t
```

If any of the node applications did not restart, use the SRVCTL utility to restart them. For example, you can use the following command to restart the `sales1` instance for the `sales` cluster database:

```
$ $ORA_CRS_HOME/bin/srvctl start instance -d sales -i "sales1"
```

12. Restart all single-instance databases that use the Oracle ASM software and all user applications that use the Oracle RAC or Oracle ASM home on each node in the group of nodes you recently patched.
13. Repeat steps 2 through 8 for the next group of nodes.
14. If you are patching nodes individually, use a command similar to the following to instruct OPatch to apply the patch to only the next node to be patched. If you run this command from the directory where the patch is located, you do not need to specify the patch ID.

    ```
    $ opatch apply -remote_nodes docrac2
    ```

    If you are patching a group of nodes, use a command similar to the following to instruct OPatch to apply the patch to the group of nodes being patched:

    ```
    $ opatch apply -remote_nodes docrac4,docrac5,docrac6
    ```

15. If you applied the patch to the Oracle Clusterware home directory, restart the CRS daemons for the nodes in the group you recently patched by issuing the following command as the `root` user on each node in the group, where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation:

    ```
    # $ORA_CRS_HOME/bin/crsctl start crs
    ```

    Repeat this command on each node in the group.

16. If you stopped the `nodeapps` for the group of nodes, after the patch has been applied, restart the `nodeapps` on those nodes. To start the `nodeapps`, enter a command similar to the following where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation and `docrac1` is one of the nodes in the group of nodes you recently patched:

    ```
    $ $ORA_CRS_HOME/bin/srvctl start nodeapps -n docrac1
    ```

    Repeat the above command for each node in the group.

    After you have restarted the `nodeapps` on each node in the group, use the `crs_stat` utility to verify that the `nodeapps` were restarted on each node in the group.

    ```
    $ $ORA_CRS_HOME/bin/crs_stat -t
    ```

    If any of the node applications did not restart, use the SRVCTL utility to restart them. For example, you can use the following command to restart the `sales2` instance for the `sales` cluster database:

    ```
    $ $ORA_CRS_HOME/bin/srvctl start instance -d sales -i "sales2"
    ```

17. Restart all single-instance databases that use the Oracle ASM software and all user applications that use the Oracle RAC or Oracle ASM home on each node in the group of nodes you recently patched.
18. If you have more than two groups of nodes to be patched, repeat steps 13 through 17 for each group of nodes until all the nodes in the cluster have been patched.
19. Run any post-patch scripts that are mentioned in the patch instructions, for example:

```
$ sqlplus /nolog
SQL> connect sys/password@sales1 AS SYSDBA
SQL> @$ORACLE_HOME/cpu/CPUOct2006/catcpu.sql
SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql
SQL> exit
```

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 up the patched nodes (the first set of nodes) before patching the remaining nodes.

To apply a patch to your cluster database using the minimum downtime method:

1. Change to the directory where the unzipped patch is staged on disk, for example:

```
$ cd $ORACLE_HOME/EMStagedPatches/4519934/4519934
```

2. Stop all user applications that use the Oracle RAC home directory for the group of nodes being patched. For example, to stop Enterprise Manager Database Control on the local node, use the following command, where `ORACLE_HOME` is the home directory for your Oracle RAC installation:

```
$ $ORACLE_HOME/bin/emctl stop dbconsole
```

3. Shut down all Oracle RAC instances on the local node. To shut down an instance for an Oracle RAC database, enter a command similar to the following example, where `ORA_CRS_HOME` is the home directory for your Oracle Clusterware

installation, `sales` is the name of the database and `sales1` is the name of the instance:

```
$ $ORA_CRS_HOME/bin/srvctl stop instance -d sales -i "sales1"
```

4. Make sure the `ORACLE_HOME` environment variable points to the software directory you want to patch, for example:

```
$ echo $ORACLE_HOME
/opt/oracle/10gR2/db_1
```

5. Use the following command from within the patch directory:

```
$ opatch apply -minimize_downtime
```

   If you run the OPatch command from the directory where the patch is staged on disk, you do not need to specify the patch ID.

   OPatch asks if you are ready to patch the local node. After you confirm that the Oracle RAC instances on the local node have been shut down, OPatch applies the patch to the Oracle RAC home directory on the local node. You are then asked to select the next nodes to be patched.

6. After you shut down the Oracle RAC instances on the other nodes in the cluster you can restart the Oracle RAC instance on the local node. Then instruct OPatch that you are ready to patch the remaining nodes.

7. After all the nodes have been patched, restart the Oracle RAC instances on the other nodes in the cluster. The following command shows how to start the `sales2` instance for the Oracle RAC database named `sales`:

```
$ $ORA_CRS_HOME/bin/srvctl start instance -d sales -i "sales1"
```

8. Verify that all the `nodeapps` were restarted on the nodes in the cluster.

```
$ crs_stat -t
```

   If any of the node applications did not restart, use the SRVCTL utility to restart them. For example, you can use commands similar the following to restart the listener on the `docrac1` node, where `ORA_CRS_HOME` is the home directory of your Oracle Clusterware installation:

```
$ $ORA_CRS_HOME/bin/srvctl start listener -n docrac1
```

9. Run any post-patch scripts that are mentioned in the patch instructions, for example:

```
$ sqlplus /nolog
SQL> connect sys/password@sales1 AS SYSDBA
SQL> @$ORACLE_HOME/cpu/CPUOct2006/catcpu.sql
SQL> @$ORACLE_HOME/rdbms/admin/utlrp.sql
SQL> exit
```

Applying Patch Sets

Patch sets are a mechanism for delivering fully tested and integrated product fixes. All of 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 server.

For instructions on applying the 10.2.0.2.0 patch set to your Oracle RAC database and Oracle Clusterware installations on Red Hat Linux, refer to document 316900.1, "ALERT: Oracle 10g release 2 (10.2) Support Status and Alerts" on the Oracle Meta*link* Web site.

This document provides a summary of the patch sets available for Oracle 10g Release 2. Using this document, you can easily locate and view the Patch Set Notes for your platform. For example, you can use a link to access document 368732.1, "Oracle Database Patch Set Notes 10g Release 2 (10.2.0.2) Patch Set for Linux x86". 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 Linux x86
- A list of known issues relating to Oracle Database on Linux x86

To locate and review the Oracle Database 10*g* Release 2 (10.2) patch set information in document 316900.1:

1. Log on to Oracle Meta*link* at `http://metalink.oracle.com`
2. Click **Advanced** at the top of the Oracle Meta*link* page.
3. Enter 316900.1 in the Document ID field, then click **Submit**.

Troubleshooting Patch Deployment

This section covers the following topics regarding troubleshooting patch deployment:

- Updating the Node List for OPatch
- Viewing 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, review these solutions to common problems. If the problem you encountered is not listed, review the log and trace files, and refer to *Oracle Universal Installer and OPatch User's Guide*.

Updating the Node List for OPatch

If OPatch does not automatically detect Oracle RAC or its nodes, investigate the contents of the inventory and ensure they are complete.

If your node list is not complete, you can update it by using the `-updateNodeList` flag of Oracle Universal Installer, as demonstrated in the following example:

```
$ORACLE_HOME/oui/bin/runInstaller -updateNodeList
ORACLE_ HOME=/opt/oracle/10gR2/db_1
CLUSTER_NODES=docrac1,docrac2,docrac3 -noClusterEnabled
```

Viewing 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 will be 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 OPatch is executed, a new log file is created.

For example, if a log file is created on May 17, 2005 at 11:55 PM, then it will be named as follows:

```
opatch_05-17-2005_23-55-00.log
```

OPatch also maintains an index of the commands executed with OPatch and the log files associated with it in the `history.txt` file located in the `$ORACLE_HOME/cfgtoollogs/opatch` directory. A sample of the `history.txt` file is as follows:

```
Date & Time : Tue Apr 26 23:00:55 PDT 2005
Oracle Home : /opt/oracle/10gR2/db_1/
OPatch Ver. : 10.2.0.0.0
Current Dir : /scratch/oui/OPatch
Command     : lsinventory
Log File    :
/opt/oracle/10gR2/db_1/cfgtoollogs/opatch/opatch-2005_Apr_26_23-00-55-PDT_Tue.log
```

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.

To correct the problem, start the OPatch utility from the directory where the patch to be installed has been unzipped and staged on disk. Or, you can use the following command when starting OPatch:

```
opatch apply /Patch_Shiphome
```

The *Patch_Shiphome* directory should have the following structure:

- An `etc` directory that has the metadata files
- A `files` directory that has the patch files
- The `etc/config/inventory` file and the actions file under the same directory

Resolving the "Unable to remove a partially installed interim patch" Error

If the patching process is interrupted, you might get the "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 execute the `restore` command as follows:

   ```
   $ORACLE_HOME/.patch_storage/patch-id_timestamp/restore.sh
   ```

3. 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.

Index

A

Adding OCR locations, 5-6
Additional Real Application Clusters
  documentation, 1-2
all node patching, 10-6
Automatic Storage Management
  file management, 1-4
  managing, 4-9
  mirroring, 1-3
  rebalancing, 1-4
  striping, 1-3
  use with Oracle RAC, 1-4
Automatic Storage Management (ASM)
  use of, 2-2

B

buffer cache, 8-4

C

Cache Fusion
  definition of, 1-3
  mechanism, 8-4
Cluster Ready Services Control (CRSCTL)
  description of, 1-5
Cluster Verification Utility (CVU)
  description of, 1-4
CRS_home
  choosing, 2-20

D

database
  backup, 6-1
  comparison of Oracle RAC to single-instance, 1-3
  recovery, 6-1
documentation
  Real Application Clusters, 1-2

E

Enterprise Manager
  and Oracle Metalink, 10-1
  backup and recovery, 6-1
  description of, 1-4
  Patch wizard, 10-2
  refreshing the patch cache, 10-2
errata level, 2-4
erratum kernel
  See also "errata level".

H

hardware requirements, 2-5
high availability framework, 5-1

I

initialization parameters
  INSTANCE_NUMBER, 4-5
  ROLLBACK_SEGMENTS, 4-5
  SERVICE_NAMES, 4-8
  that must be identical on all instances, 4-4
  that must be unique on all instances, 4-5
  THREAD, 4-5
  UNDO_MANAGEMENT, 4-5
  UNDO_TABLESPACE, 4-5
INSTANCE_NUMBER initialization parameter, 4-5
interconnects, 2-3
  definition of, 1-3

K

kernel parameters
  configuring on Linux x86 operating system, 2-15

L

Linux x86 operating system
  configuring kernel parameters, 2-15

M

membership
  and the OCR, 2-2
memory requirements, 2-5
minimum downtime patching, 10-11

N

network adapters, 2-3
Network Time Protocol (NTP), 2-14

nodes
definition of, 1-3
eviction, 5-2

O

OCR
See Oracle Cluster Registry
OPatch
"Not a valid patch area" error, 10-14
partially installed patch, 10-14
preparing to use, 10-4
updating the cluster node list, 10-13
viewing log and trace files, 10-13
Oracle Cluster Registry
definition of mirror, 2-16
defintion of, 2-2
mirroring the OCR, 5-6
multiplexing, 2-2
Oracle Clusterware
and process control, 5-1
use of, 5-1
Oracle Flashback, 6-1
Oracle Real Application Clusters
installation on different platforms, 1-2
overview of administration, 1-1, 3-1
Oracle Universal Installer
database installation, 3-1
description of, 1-4
Oracle Real Application Clusters installation, 3-1
ORACLE_BASE, 2-20
ORACLE_HOME, 3-1
OUI
See Oracle Universal Installer

P

packages, 2-4
parameters
See also initialization parameters, 4-5
that must be identical on all instances, 4-4
that must be unique on all instances, 4-5
patch sets
applying, 10-12
definition of, 10-1
Patch wizard, 10-2
patches
all node patching method, 10-6
applying, 10-5
applying using Enterprise Manager, 10-3
definition of, 10-1
locating, 10-2
minimum downtime patching, 10-11
refreshing the Enterprise Manager patch cache, 10-2
rolling patching, 10-8
staging, 10-4
physical RAM requirements, 2-5
private interconnects
and Oracle Clusterware, 2-3

R

RAM requirements, 2-5
raw devices
configuring partitions, 2-16
creating partitions, 2-17
defintion of, 2-16
required types and partition sizes for Oracle Clusterware, 2-17
Recovery Manager (RMAN)
channels, 6-5
parallelism and backups, 6-5
redundancy
normal, 5-1
requirements
hardware, 2-5
ROLLBACK_SEGMENTS initialization parameter, 4-5
rolling patch method, 10-8

S

Server Control (SRVCTL)
description of, 1-4
SERVICE_NAMES initialization parameter, 4-8
services
available instance, 7-2
preferred instance, 7-2
shared storage
supported types, 2-15
slices
See also raw devices
creating partitions, 2-17
software requirements
checking software requirements, 2-4
swap space
requirements, 2-5
System Global Area (SGA), 8-4

T

temporary disk space
requirements, 2-5
THREAD initialization parameter, 4-5

U

UNDO_MANAGEMENT initialization parameter, 4-5
UNDO_TABLESPACE initialization parameter, 4-5
user equivalence, 2-9, 10-5

V

voting disks
definition of, 2-2
multiple, 2-2