

Oracle® Real Application Clusters

Administrator's Guide

10g Release 1 (10.1)

Part No. B10765-01

December 2003

Oracle Real Application Clusters Administrator's Guide, 10g Release 1 (10.1)

Part No. B10765-01

Copyright © 1998, 2003, Oracle. All rights reserved.

Primary Author: David Austin and Mark Bauer.

Contributor: Jonathan Creighton, Rajiv Jayaraman, Raj Kumar, Dayong Liu, Venkat Maddali, Michael Salinas, Sudheendra Sampath, Cathy Shea, and Khethavath P Singh.

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 Corporation, 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 is a registered trademark 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

Send Us Your Comments	ix
Preface	xi
Intended Audience.....	xii
Documentation Accessibility	xii
Structure	xii
Related Documents	xiii
Conventions	xiv
What's New in Administration?	xix
Oracle Database 10g New Features for Real Application Clusters Administration.....	xx
1 Introduction to Administration	
Introduction to Administering Real Application Clusters	1-2
Administering Real Application Clusters	1-2
Database Instance Management and Database Administration in RAC	1-2
Storage Management in Real Application Clusters	1-2
Administering Services in Real Application Clusters	1-2
Additional Real Application Clusters Administrative Topics	1-3
Overview of Using Enterprise Manager with Real Application Clusters	1-3
2 Administering Database Instances in Cluster Databases	
Database Component Overview	2-2
Overview of Real Application Clusters Management Tools	2-2
Overview of Administering Real Application Clusters with Enterprise Manager.....	2-2
Overview of Administering Real Application Clusters with SQL*Plus	2-2
Overview of Administering Real Application Clusters with SRVCTL.....	2-3
Starting and Stopping Instances and Real Application Clusters Databases	2-3
Starting up and Shutting down with Enterprise Manager	2-4
Starting up and Shutting down with SQL*Plus	2-4
Starting up and Shutting down with SRVCTL	2-5
Overview of Initialization Parameter Files in Real Application Clusters	2-5
Setting Server Parameter File Parameter Values for Real Application Clusters	2-6
Exporting the Server Parameter File for Backward Compatibility	2-7

Initialization Parameter Use in Real Application Clusters	2-7
Parameters that Must Have Identical Settings on All Instances	2-7
Parameters That Must Have Unique Settings on All Instances	2-8
Summary of Parameter Use in Real Application Clusters Databases	2-8
Backing Up the Server Parameter File	2-10

3 Administering Storage

Overview of Storage in Oracle Real Application Clusters	3-2
Automatic Storage Management	3-2
Automatic Storage Management Components in RAC	3-2
Modifying Disk Group Configurations for ASM in RAC	3-2
Administering ASM Instances and ASM Disk Groups with Enterprise Manager in RAC.....	3-3
Administering ASM Instances with SRVCTL in RAC	3-3
Datafile Access in Real Application Clusters.....	3-4
Redo Log File Storage in Real Application Clusters.....	3-4
Automatic Undo Management in Real Application Clusters.....	3-4
Switching Undo Tablespaces for Instances in Real Application Clusters	3-5
Managing the Oracle Cluster Registry in Real Application Clusters	3-5
Managing the Oracle Cluster Registry with OCR Exports	3-6
The ocrconfig Tool Command Syntax and Options.....	3-7
Implementing the Oracle Hardware Assisted Resilient Data Initiative for the OCR	3-8
Upgrading and Downgrading the OCR Configuration in Real Application Clusters	3-8
Importing and Exporting Cluster Database Configuration Information with SRVCONFIG ...	3-8

4 Administering Services

Services and High Availability in Real Application Clusters	4-2
Adding and Modifying Services	4-2
Using the DBCA to Add and Modify Services	4-3
Automatic Restarts after Failures	4-3
Administering Services with Enterprise Manager and SRVCTL	4-4
Administering Services in Real Application Clusters with Enterprise Manager	4-4
Administering Services in Real Application Clusters with SRVCTL	4-5

5 Adding and Deleting Nodes and Instances

Overview of Node Addition Procedures	5-2
Step 1: Connecting New Nodes to the Cluster	5-2
Making Physical Connections	5-2
Installing Operating System	5-3
Creating Oracle Users.....	5-3
Checking the Installation	5-3
Step 2: Extending Clusterware and Oracle Software to New Nodes	5-3
Adding Nodes at the Vendor Clusterware Layer (UNIX only)	5-4
Adding Nodes at the Oracle Clusterware Layer (UNIX and Windows).....	5-4
Step 3: Preparing Storage for RAC on New Nodes	5-6
Raw Device Storage Preparation for New Nodes	5-7
Step 4: Adding Nodes at the Oracle RAC Database Layer	5-8

Step 5: Adding Database Instances to New Nodes	5-10
Adding Nodes that Already Have Clusterware and Oracle Software to a Cluster	5-12
Deleting Instances from Real Application Clusters Databases	5-12
Deleting Nodes from Oracle Clusters on UNIX-Based Systems	5-13
Deleting Nodes from Oracle Clusters on Windows-Based Platforms	5-14
6 Configuring Recovery Manager and Archiving	
Overview of Configuring RMAN for Real Application Clusters.....	6-2
Configuring the RMAN Snapshot Control File Location	6-2
Configuring the RMAN Control File Autobackup Feature	6-2
Managing Archived Redo Logs Using RMAN in Real Application Clusters.....	6-3
Archived Redo Log File and Destination Conventions in RAC	6-4
RMAN Archiving Configuration Scenarios.....	6-4
Cluster File System Archiving Scheme	6-5
Non-Cluster File System Local Archiving Scheme	6-6
Changing the Archiving Mode in Real Application Clusters	6-8
Monitoring the Archiver Processes	6-8
7 Managing Backup and Recovery	
Instance Recovery in Real Application Clusters.....	7-2
Single Node Failure in Real Application Clusters.....	7-2
Multiple-Node Failures in Real Application Clusters	7-2
Using RMAN to Create Backups in Real Application Clusters	7-2
Using RMAN to Restore the Server Parameter File (SPFILE)	7-2
RMAN and Oracle Net in Real Application Clusters	7-3
Channel Connections to Cluster Instances.....	7-3
Node Affinity Awareness of Fast Connections	7-3
Readability of Files To Be Backed Up	7-3
Deleting Archived Redo Logs After a Successful Backup	7-4
Distribution of Backups.....	7-4
Autolocation for Backup and Restore Commands.....	7-4
RMAN Backup Scenarios for Real Application Clusters.....	7-5
Cluster File System Backup Scheme.....	7-5
Non-Cluster File System Backup Scheme.....	7-6
Media Recovery in Real Application Clusters	7-6
RMAN Restore Scenarios for Real Application Clusters.....	7-7
Cluster File System Restore Scheme.....	7-7
Non-Cluster File System Restore Scheme.....	7-7
RMAN Recovery Through Resetlogs in Real Application Clusters	7-8
Parallel Recovery in Real Application Clusters	7-8
Parallel Recovery with RMAN.....	7-8
Parallel Recovery with SQL*Plus.....	7-8
Using a Flash Recovery Area in Real Application Clusters.....	7-9
8 Administrative Options	
Optional Enterprise Manager Tasks in Real Application Clusters	8-2

Using Enterprise Manager to Discover Nodes and Instances	8-2
Enterprise Manager Pages for Real Application Clusters.....	8-2
Databases Summary Page.....	8-3
Cluster Database Home Page.....	8-3
Cluster Database Instances Pages.....	8-3
The Databases Overview Page for Real Application Clusters	8-3
The Cluster Home Page for Real Application Clusters	8-3
Instance Pages for Real Application Clusters.....	8-4
Real Application Clusters Administration Procedures for Enterprise Manager.....	8-4
Administering Enterprise Manager Jobs in Real Application Clusters	8-4
Administering Alerts in Real Application Clusters with Enterprise Manager.....	8-5
Performing Scheduled Maintenance Using Defined Blackouts in Enterprise Manager	8-5
Additional Information About SQL*Plus in Real Application Clusters	8-5
How SQL*Plus Commands Affect Instances	8-5
Verifying that Instances are Running.....	8-6
Quiescing Real Application Clusters Databases	8-7
Quiesced State and Cold Backups	8-7
Administering System and Network Interfaces with the OIFCFG (Oracle Interface Configuration) Tool	8-7
Defining Network Interfaces with OIFCFG	8-8
Syntax and Commands for the OIFCFG Command-Line Tool.....	8-8

A Troubleshooting

Monitoring Trace Files in Real Application Clusters.....	A-2
Where to Find Files for Analyzing Errors	A-2
Using Log Files in Real Application Clusters.....	A-2
Clusterware Log Files	A-2
Enabling Additional Tracing for Real Application Clusters High Availability.....	A-3
Using Instance-Specific Alert Files in Real Application Clusters.....	A-5
Resolving Pending Shutdown Issues	A-5

B Server Control (SRVCTL) Reference

Overview of SRVCTL for Administering Real Application Clusters.....	B-2
Guidelines for Using SRVCTL in Real Application Clusters.....	B-2
Obtaining Command Line Help for SRVCTL.....	B-2
SRVCTL Command Syntax and Options	B-2
SRVCTL Cluster Database Configuration Tasks	B-3
SRVCTL General Cluster Database Administration Tasks	B-3
SRVCTL Node-Level Tasks	B-3
SRVCTL Command Reference	B-4
SRVCTL Commands.....	B-4
SRVCTL Commands Summary	B-4
SRVCTL Objects Summary	B-5
srvctl add	B-5
srvctl config.....	B-8
srvctl enable	B-10
srvctl disable	B-12

srvctl start	B-14
srvctl stop	B-17
srvctl modify	B-20
srvctl relocate	B-23
srvctl status	B-24
srvctl getenv	B-26
srvctl setenv and unsetenv.....	B-28
srvctl remove.....	B-32

C Oracle Real Application Clusters Tools Messages

Overview of Real Application Clusters-Specific Messages	C-2
Prefixes and Message Codes for RAC-Specific Messages	C-2
Types of Real Application Clusters Messages and Related Files	C-2
PRKA—Cluster Node Applications Messages	C-2
PRKC—Cluster Command Messages.....	C-2
PRKD — Global Services Daemon Messages	C-8
PRKE — Global Services Daemon Controller Utility Messages	C-9
PRKH—Server Manager (SRVM) Messages.....	C-9
PRKN— Server Manager (SRVM) System Library Messages.....	C-11
PRKO—Server Control (SRVCTL) Utility Messages	C-11
PRKP—Cluster Database Management Messages	C-15
PRKR—Cluster Registry Messages	C-22
PRKS—Automatic Storage Management Messages.....	C-27
PRKU—Command Line Parser Utility Messages	C-31
PRKV — Virtual IP Configuration Assistant Messages	C-31

Index

Send Us Your Comments

Oracle Real Application Clusters Administrator's Guide, 10g Release 1 (10.1)

Part No. B10765-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227. Attn: Server Technologies Documentation Manager
- Postal service:

Oracle Corporation
Server Technologies Documentation Manager
Street Address, Mailstop 4op11
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

The *Oracle Real Application Clusters Administrator's Guide* describes the administrative tasks specific to Oracle Real Application Clusters (RAC). Information in this manual applies to RAC as it runs on all platforms. In addition, the content of this manual supplements administrative content for Oracle single-instance databases in other Oracle documentation. Where necessary, this manual refers to platform-specific documentation. This preface contains these topics:

This Preface contains these topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Structure](#)
- [Related Documents](#)
- [Conventions](#)

Intended Audience

The *Oracle Real Application Clusters Administrator's Guide* is intended for database administrators, network administrators, and system administrators who perform the following tasks:

- Administer and manage RAC databases
- Manage and troubleshoot clusters and networks that use RAC

To use this document, you should be familiar with the administrative procedures described in *Oracle 2 Day DBA* and the *Oracle Database Administrator's Guide*. You should also be familiar with installing and configuring RAC as described in *Oracle Real Application Clusters Installation and Configuration Guide*.

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. 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 additional information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, 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, JAWS 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.

Structure

This document is organized as follows:

Chapter 1, "Introduction to Administration"

This chapter introduces the administrative tasks for RAC software.

Chapter 2, "Administering Database Instances in Cluster Databases"

This chapter explains how to administer RAC databases and instances

Chapter 3, "Administering Storage"

This chapter explains how to administer storage components in RAC.

Chapter 4, "Administering Services"

This chapter describes how to administer services in RAC environments.

Chapter 5, "Adding and Deleting Nodes and Instances"

This chapter explains the procedures for adding and deleting nodes and instance in RAC database environments.

Chapter 6, "Configuring Recovery Manager and Archiving"

This chapter explains how to configure Recovery Manager (RMAN) for use with RAC databases.

Chapter 7, "Managing Backup and Recovery"

This chapter explains how to administer backup and recovery in RAC.

Chapter 8, "Administrative Options"

This chapter describes administrative options for Real Application Clusters.

Appendix A, "Troubleshooting"

This appendix explains how to contact Oracle Support Services.

Appendix B, "Server Control (SRVCTL) Reference"

This appendix is an SRVCTL command reference.

Appendix C, "Oracle Real Application Clusters Tools Messages"

This appendix describes the messages for RAC management tools.

Related Documents

For more information, refer to the Oracle resources listed in this section.

- *Oracle Real Application Clusters Installation and Configuration Guide*
- *Oracle Real Application Clusters Deployment and Performance Guide*
- *Oracle Database Administrator's Guide*
- *Oracle 2 Day DBA*
- *Oracle Net Services Administrator's Guide*
- *Oracle Database Platform Guide for Windows*
- *Oracle Database 10g Administrator's Reference Release 1 (10.1) for UNIX Systems: AIX-Based Systems, HP-UX, Tru64 UNIX, Linux, and the Solaris Operating System (SPARC)*

Error messages are only available online or by way of a Tahiti documentation search.

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

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

<http://otn.oracle.com/membership/>

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

<http://otn.oracle.com/documentation/>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to start SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>old_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt"\system32 is the same as C:\WINNT\SYSTEM32
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.	C:\oracle\oradata>
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
<i>HOME_NAME</i>	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	C:\> net start Oracle <i>HOME_NAME</i> TNSListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\orann, where <i>nn</i> is the latest release number. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle Database Platform Guide for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

What's New in Administration?

This section describes the new features for Oracle Database 10g Release 1 (10.1) for Oracle Real Application Clusters (RAC) administration. The topic in this section is:

- [Oracle Database 10g New Features for Real Application Clusters Administration](#)

See Also:

- *Oracle Real Application Clusters Installation and Configuration Guide*
- *Oracle Real Application Clusters Deployment and Performance Guide*
- *Oracle Database New Features*
- *Oracle Enterprise Manager Grid Control Installation and Basic Configuration*

Oracle Database 10g New Features for Real Application Clusters Administration

This section describes the Oracle Database 10g features for RAC administration.

- High Availability, Workload Management, and Services

Oracle Real Application Clusters introduces integrated clusterware known as Cluster Ready Services (CRS). You install CRS on all platforms on which you can run Oracle Real Application Clusters software. CRS manages cluster database functions including node membership, group services, global resource management, and high availability.

In Oracle Real Application Clusters, you can use services to define application workloads by creating a service for each application or for major components within complex applications. You can then define where and when the service runs and thus use services to control your workload.

In both cluster and non-cluster environments, the Automatic Workload Repository (AWR) tracks performance metrics using services. You can also set thresholds on performance metrics to automatically generate alerts if these thresholds are exceeded.

See Also: *PL/SQL Packages and Types Reference* for more information about the `DBMS_SERVICE` PL/SQL and `DBMS_MONITOR` packages and for more information about setting thresholds.

- Enhanced Cluster Manager Implementation

In earlier releases of the Oracle Database, cluster manager implementations on some platforms were referred to as "Cluster Manager". In Oracle Database 10g, Cluster Ready Services (CRS) serves as the clusterware software, and Cluster Synchronization Services (CSS) is the cluster manager software for all platforms. The Oracle Cluster Synchronization Service Daemon (OCSSD) performs some of the clusterware functions on UNIX-based systems. On Windows-based systems, OracleCSService, OracleCRService, and OracleEVMSservice replace the Oracle Database OracleCMService*9i*.

Oracle Enterprise Manager, the Database Configuration Assistant (DBCA), and the Server Control (SRVCTL) Utility provide tools to administer clusters, RAC databases, and services.

- Enterprise Manager Enhancements for RAC

This release includes the new Web-based Enterprise Manager Database Control with which you can manage a RAC database, and Enterprise Manager Grid Control for administering multiple RAC databases. Administration of RAC databases is greatly simplified because of more simplified drill-down tasks and because Enterprise Manager displays cluster-wide performance information. This is available for both single-instance Oracle and RAC databases.

Enterprise Manager has several summary pages that show cluster database performance information at a glance; you no longer have to log in to each cluster database or display instance-specific pages to obtain a global view of cluster database performance.

- Enhancements for Flash Recovery Area and Automatic Disk-Based Backup and Recovery

- A flash recovery area is an Automatic Storage Management (ASM) disk group, a file system, or a directory that serves as a default storage area for recovery files. RAC supports the Automatic Disk-Based Backup and Recovery feature that simplifies managing disk space and backup and recovery files.
- Database Configuration Assistant (DBCA) Enhancements
- Use the DBCA to perform instance addition and deletion as well as database deletion.
- Database Upgrade Assistant (DBUA) Enhancements
Use the DBUA to upgrade from an earlier RAC version to Oracle Database 10g with RAC. When you upgrade from a Primary/Secondary environment, the DBUA creates one service and assigns it to one instance as a preferred instance, and to the other instance as its available instance.

Server Control (SRVCTL) Enhancements

Enhancements to SRVCTL support the management of services and Automatic Storage Management (ASM) instances within RAC.

- Enhanced Recovery Parallelism on Multiple CPU Systems
The default for instance, crash, and media recovery is to operate in parallel mode on multiple-CPU systems.
- Revised Error Messages for High Availability and Management Tools in Real Application Clusters
- The high availability error messages have been enhanced for this release.
- Oracle Cluster Registry (OCR) Enhancements
The OCR contains configuration details for the cluster database and for high availability resources such as services, Virtual Interconnect Protocol (VIP) addresses, and so on.
- GCS_SERVER_PROCESSES Parameter
There is a new, static parameter to specify the number of server processes for an instance's Global Cache Service (GCS) for routing inter-instance traffic among RAC instances. The default number of GCS server processes is calculated based on system resources with a minimum of 2. You can set this parameter to different values on different instances.

Introduction to Administration

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

- [Introduction to Administering Real Application Clusters](#)
- [Administering Real Application Clusters](#)
- [Database Instance Management and Database Administration in RAC](#)

Introduction to Administering Real Application Clusters

Install your Oracle Database 10g software with the Oracle Universal Installer (OUI) and create your database with the Database Configuration Assistant (DBCA). This ensures that your RAC environment has the optimal network configuration, database structure, and parameter settings for the environment that you selected. As a DBA, after installation your tasks are to administer your RAC environment at three levels:

- Instance Administration
- Database Administration
- Cluster Administration

Administering Real Application Clusters

Use the following tools to perform administrative tasks in RAC:

- Enterprise Manager—Oracle recommends that you use Enterprise Manager to perform administrative tasks whenever feasible
- Task-specific GUIs such as the Database Configuration Assistant (DBCA) and the Virtual Internet Protocol Configuration Assistant (VIPCA)
- Command-line tools such as SQL*Plus, Server Control (SRVCTL), and the Oracle Interface Configuration tool (OIFCFG)

Database Instance Management and Database Administration in RAC

Use Enterprise Manager, SQL*Plus, or SRVCTL to administer database instances and RAC databases as described in [Chapter 2, "Administering Database Instances in Cluster Databases"](#).

Storage Management in Real Application Clusters

When you create your database, you can create Automatic Storage Management (ASM) disk groups and configure mirroring for ASM disk groups using the DBCA. After your RAC database is operational, you can administer ASM disk groups with Enterprise Manager or the SRVCTL utility as described in [Chapter 3, "Administering Storage"](#).

Administering Services in Real Application Clusters

When you create a RAC database, you can also create services and assign them to instances using the DBCA. After your RAC database is operational, you can use the DBCA, as well as Enterprise Manager and SRVCTL to administer services and high availability components as described in [Chapter 4, "Administering Services"](#).

Other high availability components include node resources such as the Virtual Internet Protocol (VIP) address for each node, the Global Services Daemon, the Enterprise Manager Agent, and the Oracle Net Listeners. These resources are automatically started when Cluster Ready Services (CRS) starts the node and CRS automatically restarts them if they fail. The application level resources are the instances and CRS background processes that run on each instance.

You can use the VIPCA to administer VIP addresses, and SRVCTL to administer other node resources. The information that describes the configuration of these components

is stored in the Oracle Cluster Registry (OCR) which you administer as described in [Chapter 3, "Administering Storage"](#).

See Also: *Oracle Real Application Clusters Deployment and Performance Guide* for more information about CRS

Additional Real Application Clusters Administrative Topics

This book contains the following additional RAC administrative topics:

- Scalability—Adding instances and nodes to a RAC database as described in [Chapter 5, "Adding and Deleting Nodes and Instances"](#)
- Backup and Recovery—Configuring Recovery Manager (RMAN) and performing backup and recovery processing as described in [Chapter 6, "Configuring Recovery Manager and Archiving"](#) and [Chapter 7, "Managing Backup and Recovery"](#)
- Log Files—Administering information that Oracle records in log files as described in [Appendix A, "Troubleshooting"](#) and [Appendix B, "Server Control \(SRVCTL\) Reference"](#)
- Using SRVCTL—Using SRVCTL to administer RAC instances, databases, services, and so on, as described in [Appendix B, "Server Control \(SRVCTL\) Reference"](#)
- Error Messages—Interpreting error messages for RAC high availability and management tools as described in [Appendix C, "Oracle Real Application Clusters Tools Messages"](#)

Overview of Using Enterprise Manager with Real Application Clusters

Enterprise Manager is a Web-based tool with RAC-specific administration and performance-related features. If you create your RAC database with the Database Configuration Assistant (DBCA), then the Enterprise Manager Database Control tool is automatically configured for your RAC environment. This means that all instances that were part of your installation have an Enterprise Manager Agent running on them. Enterprise Manager Database Control enables you to manage a single RAC database with its instance targets, Oracle Net Services listener targets, host targets, and a cluster target.

Additionally, you can configure Enterprise Manager Grid Control on other hosts either inside or outside your cluster environment. Enterprise Manager Grid Control enables you to manage multiple cluster databases, cluster database instances, and the hosts on which cluster database components operate.

Grid Control enables you to monitor and administer your entire computing environment from one network location. Use Grid Control to manage all of your enterprise services, including hosts, databases, listeners, application servers, HTTP Servers, and Web applications, as one cohesive unit. Enterprise Manager Grid Control only requires one Agent on one host in your cluster environment to perform cluster database and instance discovery. Install Enterprise Manager Grid Control from a separate CD-ROM that is part of the Oracle Database 10g Release 1 (10.1) CD-ROM Pack.

See Also: *Oracle Enterprise Manager Concepts* for more information about using Enterprise Manager

You can also use both Enterprise Manager Database Control and the Enterprise Manager Grid Control to:

- Administer database services—Start, stop, relocate, obtain status, and so on
- Create and assign resource plans—Assign resource plans to cluster database instances
- Administer storage—Assign undo tablespaces and re-assign them from one instance to another, administer redo log assignments among cluster database instances, and switch archive log modes
- Administer Automatic Storage Management—Administer ASM instances and ASM disk groups
- Perform general database activities—Start up and shut down RAC databases and instances, perform backup and recovery operations, edit server parameter file (spfile) settings for instances or for entire cluster databases, and so on
- Display host configurations—Memory, CPU, device I/O, network interfaces, the operating system and installed patches

Administering Database Instances in Cluster Databases

This chapter describes how to administer Oracle Real Application Clusters (RAC) database instances and RAC databases. This chapter explains the startup and shutdown tasks for database components and well as how to administer parameters and parameter files in RAC. The topics in this chapter are:

- [Database Component Overview](#)
- [Overview of Real Application Clusters Management Tools](#)
- [Starting and Stopping Instances and Real Application Clusters Databases](#)
- [Overview of Initialization Parameter Files in Real Application Clusters](#)
- [Initialization Parameter Use in Real Application Clusters](#)
- [Summary of Parameter Use in Real Application Clusters Databases](#)
- [Backing Up the Server Parameter File](#)

See Also: [Chapter 3, "Administering Storage"](#) for information about managing Automatic Storage Management (ASM) instances

Database Component Overview

RAC databases comprise a control file, redo logs, datafiles, and one server parameter file (SPFILE) or one or more PFILEs, or client-side parameter files. The instances comprise the System Global Area (SGA) and the instance background processes.

Overview of Real Application Clusters Management Tools

The following section introduces the three tools you will most likely to use to manage an existing RAC database and its instances, Oracle Enterprise Manager, SQL*Plus, and the SRVCTL utility. In many cases, you use these tools the same way to manage a single-instance database but the following sections identify the important differences when managing a RAC database:

- [Overview of Administering Real Application Clusters with Enterprise Manager](#)
- [Overview of Administering Real Application Clusters with SQL*Plus](#)
- [Overview of Administering Real Application Clusters with SRVCTL](#)

Overview of Administering Real Application Clusters with Enterprise Manager

Use the Web-based Enterprise Manager Database Control to manage a single RAC database. The Enterprise Manager Console provides a central point of control for the Oracle environment through a graphical user interface (GUI). You can use the Enterprise Manager Console to initiate a variety of cluster database management tasks. Use Enterprise Manager Grid Control to administer multiple RAC databases.

Enterprise Manager enables you to start, stop, and monitor databases, cluster database instances, and their listeners, as well as to schedule jobs or register events. You can perform these tasks simultaneously on multiple cluster databases. You can also use the Console to manage schemas, security, and cluster database storage features.

Overview of Administering Real Application Clusters with SQL*Plus

SQL*Plus commands operate on the current instance. The current instance can be either the local default instance on which you initiated your SQL*Plus session, or it can be a remote instance to which you connect with Net Services.

Because the SQL*Plus prompt does not display the current instance, you should direct your commands to the correct instance. Starting a SQL*Plus session and connecting to the database without specifying an instance directs all SQL*Plus commands to the local instance. In this case, the default instance is also the current instance.

To connect to a different instance in SQL*Plus, issue a new CONNECT command specify a remote instance net service name as in the following example:

```
CONNECT <user name>/<password>@net_service_name
```

Connecting as SYSOPER or SYSDBA enables you to perform privileged operations, such as instance startup and shutdown. Multiple SQL*Plus sessions can connect to the same instance at the same time. SQL*Plus automatically disconnects you from the first instance whenever you connect to another one.

See Also:

- *Oracle Net Services Administrator's Guide* for the proper specification of `net_service_name`
- The *Oracle Database Administrator's Guide* for information about connecting to the database using `SYSDBA` or `SYSOPER` privileges

Overview of Administering Real Application Clusters with SRVCTL

The SRVCTL tool manages configuration information that is used by several other Oracle tools. For example, Enterprise Manager uses the configuration information that SRVCTL generates to discover and monitor nodes in your cluster.

When you use SRVCTL to perform configuration operations on your cluster, SRVCTL stores configuration data in the Server Management (SRVM) configuration repository. SRVCTL performs other operations, such as starting and stopping instances, by calling SQL*Plus on each node. SRVCTL uses the same Oracle Cluster Registry (OCR) that is used with other Oracle administrative interfaces.

Starting and Stopping Instances and Real Application Clusters Databases

You can start up and shut down instances with Enterprise Manager, SQL*Plus or SRVCTL as described in the following sections. Both Enterprise Manager and SRVCTL provide options to startup and shutdown all of the instances in a RAC database with a single step.

You can only perform certain operations when the database is in a NOMOUNT or MOUNT state. Performing other operations requires that the database be OPEN. In addition, some operations require that only one instance be in the required state, while other operations require that all of the instances be in an identical state.

The procedures in this section assume that you are using a server parameter file (SPFILE) and are described in the following topics:

- [Starting up and Shutting down with Enterprise Manager](#)
- [Starting up and Shutting down with SQL*Plus](#)
- [Starting up and Shutting down with SRVCTL](#)

Before you can start a RAC instance your clusterware and any required operating system-specific processes. For more information about these processes, see your operating system documentation.

The procedure for shutting down RAC instances is identical to shutting down instances in single-instance Oracle, with the exceptions described here. Refer to the *Oracle Database Administrator's Guide* for more information about shutting down Oracle databases.

- In RAC, shutting down one instance does not interfere with the operation of other running instances.
- To shut down a RAC database mounted in shared mode, shut down every instance in the RAC environment.
- After a NORMAL or IMMEDIATE shutdown, instance recovery is not required. Recovery is required, however, after you issue the SHUTDOWN ABORT command or after an instance terminates abnormally. The instance that is still running performs instance recovery for the instance that shut down. If no other instances are

running, the next instance to open the database performs instance recovery for any instances needing it.

- The `SHUTDOWN TRANSACTIONAL` command with the `LOCAL` option is useful to shutdown an instance after all active transactions on the instance have either committed or rolled back. This is in addition to what this command does for `SHUTDOWN IMMEDIATE`. Transactions on other instances do not block this operation. If you omit the `LOCAL` option, then this operation waits until transactions on all other instances that started before the shutdown was issued either commit or rollback.

Starting up and Shutting down with Enterprise Manager

To access a cluster database instance, from the Home page, click the Targets tab, then click the cluster database name. On the Cluster Database Home page, the cluster database instances display at the bottom of the page. Click an instance name to go to the Cluster Database Instance Home page where you can start or stop the cluster database instance as well as see an overview of the cluster database instance activity such as CPU and space usage, active sessions, and so on.

To start a cluster database instance click Startup, or click Shutdown to stop it. To start or shutdown a cluster database, that is, all of the instances known to Enterprise Manager, select the database and click Startup or Shutdown on the Cluster Database page.

Starting up and Shutting down with SQL*Plus

If you want to start or stop just one instance and you are connected to your local node, you should first ensure that your current environment includes the SID for the local instance. Note that any subsequent commands in your session, whether inside or outside a SQL*Plus session, will be associated with that same SID.

To start or shutdown your local instance, initiate a SQL*Plus session and connect with the `SYSDBA` or `SYSOPER` privilege and then issue the required command. For example to start and mount an instance on your local node, execute the following commands within your SQL*Plus session:

```
CONNECT / AS SYSDBA
STARTUP MOUNT
```

You can start multiple instances from a single SQL*Plus session on one node by way of Oracle Net Services. To achieve this, you must connect to each instance in turn by using a Net Services connection string, typically an instance-specific alias from your `TNSNAMES.ORA` file.

Note: To ensure that you connect to the correct instance, you must use an alias in the connect string that is associated with just one instance. If you use an alias to a service or with multiple addresses, you may not be connected to your intended instance.

For example, you can use a SQL*Plus session on a local node to perform a transactional shutdown for two instances on remote nodes by connecting to each in turn using the instance's individual alias name. Assume the alias name for the first instance is `db1` and that the alias for the second instance is `db2`. Connect to the first instance and shut it down as follows:

```
CONNECT /@db1 AS SYSDBA
```

```
SHUTDOWN TRANSACTIONAL
```

Then connect to and shutdown the second instance by entering the following from your SQL*Plus session:

```
CONNECT /@db2 AS SYSDBA
SHUTDOWN TRANSACTIONAL
```

Other startup and shut down keywords, such as NOMOUNT, MOUNT, IMMEDIATE, and so on, are described in the *SQL*Plus User's Guide and Reference*.

It is not possible to start up or shut down more than one instance at a time in SQL*Plus, so you cannot start or stop all of the instances for a cluster database with a single SQL*Plus command. You may wish to create a script that will connect to each instance in turn and start it up and shut it down. However, you will need to maintain this script manually if you add or drop instances.

Starting up and Shutting down with SRVCTL

Enter the following SRVCTL syntax from the command line, providing the required database name and instance name, or include multiple instance names to start more than one specific instance:

```
srvctl start instance -d <db_name> -i <inst_name_list> [-o <start_options>] [-c
<connect_str> | -q]
```

Note that this command will also start all enabled and non-running services that have the listed instances either as preferred or available instances.

To stop one or more instances, enter the following SRVCTL syntax from the command line:

```
srvctl stop instance -d <name> -i <inst_name_list> [-o <stop_options>] [-c
<connect_str> | -q]
```

This command will also stop the services related to the terminated instances on the nodes where the instances were running.

To start or stop your entire cluster database, that is, all of the instances and its enabled services, enter the following SRVCTL commands:

```
srvctl start database -d <name> [-o <stop_options>] [-c <connect_str> | -q]
```

```
srvctl stop database -d <name> [-o <stop_options>] [-c <connect_str> | -q]
```

See Also: [Appendix B, "Server Control \(SRVCTL\) Reference"](#) for information about SRVCTL options and information about other administrative tasks that you can perform with SRVCTL

Overview of Initialization Parameter Files in 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. If you manually create your database, then Oracle recommends that you create an SPFILE from an initialization parameter file (PFILE).

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

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 file 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.

Setting Server Parameter File Parameter Values for Real Application Clusters

You can alter SPFILE settings with Enterprise Manager or by using the SET clause of the ALTER SYSTEM statement. In addition, the ALTER SYSTEM syntax enables you to override the effects of SPFILE settings that you make manually. However, if your SPFILE contains instance-specific settings, then these settings take precedence over settings made with ALTER SYSTEM commands.

The examples in this section appear in ASCII text although the SPFILE is a binary file. Assume that you start an instance with an SPFILE containing the following entries:

```
*.OPEN_CURSORS=500
prod1.OPEN_CURSORS=1000
```

For the instance with the Oracle system identifier (sid) *prod1*, the OPEN_CURSORS parameter remains set to 1000 even though it has a database-wide setting of 500. The instance-specific parameter setting in the parameter file for an instance prevents database-wide alterations of the setting. This gives you control over parameter settings for instance *prod1*. These two types of settings can appear in any order in the parameter file.

If another DBA runs the following statement, then Oracle updates the setting on all instances except the instance with sid *prod1*:

```
ALTER SYSTEM SET OPEN_CURSORS=1500 sid='' SCOPE=MEMORY;
```

In the example instance with sid *prod1*, the parameter begins accepting ALTER SYSTEM values set by other instances if you change the parameter setting by running the following statement:

```
ALTER SYSTEM RESET OPEN_CURSORS SCOPE=MEMORY sid='prod1';
```

Then if you execute the following statement on another instance, the instance with sid *prod1* also assumes the new setting of 2000:

```
ALTER SYSTEM SET OPEN_CURSORS=2000 sid='' SCOPE=MEMORY;
```

In the following example, the server parameter file contains these entries:

```
prod1.OPEN_CURSORS=1000
*.OPEN_CURSORS=500
```

Running the following statement makes Oracle disregard the first entry from the server parameter file:

```
ALTER SYSTEM RESET OPEN_CURSORS SCOPE=SPFILE sid='prod1';
```

To reset a parameter to its default value throughout your cluster database, enter the statement:

```
ALTER SYSTEM RESET OPEN_CURSORS SCOPE=SPFILE sid='*';
```

Note: Modifying SPIFLEs with anything except Enterprise Manager or SQL*Plus can corrupt the file and prevent database startup. To repair the file, you might need to create the PFILE and regenerate the SPFILE.

Specifying Comments in Server Parameter Files

Specify comments on the same line with the parameter setting. For example, if the initialization parameter file contains the following lines:

```
# first comment
OPEN_CURSORS = value # second comment
```

The string `second comment` is associated with the setting for `OPEN_CURSORS`. Oracle displays this comment in the `V$PARAMETER` and `V$PARAMETER2` views. Oracle also displays comments such as the entry `#first comment` in the example.

Exporting the Server Parameter File for Backward Compatibility

You can revert to previous releases of RAC and convert from using the server parameter file to the traditional client-side PFILE. The procedure for this is described in *Oracle Database Administrator's Guide*.

Initialization Parameter Use in Real Application Clusters

Most initialization parameters can have different values on different instances as described in the *Oracle Database Reference*. A parameter that can optionally have a different value for each instance has a default value that is typically the same on all instances. You can change the value on one or more instances and your new values can be unique across your instances or the values can be the same on one or more of your instances. Other parameters must either be identical or unique as described in the following sections.

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 RAC. Specify these parameter values in the SPFILE, or within each `init_dbname.ora` file on each instance. The following list contains the parameters that must be identical on every instance:

- ACTIVE_INSTANCE_COUNT
- ARCHIVE_LAG_TARGET
- CLUSTER_DATABASE
- CONTROL_FILES
- DB_BLOCK_SIZE
- DB_DOMAIN
- DB_FILES
- DB_NAME
- DB_RECOVERY_FILE_DEST
- DB_RECOVERY_FILE_DEST_SIZE

- DB_UNIQUE_NAME
- MAX_COMMIT_PROPAGATION_DELAY
- TRACE_ENABLED
- 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

If you use the THREAD or ROLLBACK_SEGMENTS parameters, then Oracle recommends setting unique values for them by using the sid identifier in the SPFILE. However, you must set a unique value for INSTANCE_NUMBER for each instance and you cannot use a default value.

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

Specify the ORACLE_SID environment variable, which comprises the database name and the number of the THREAD assigned to the instance.

If you specify UNDO_TABLESPACE with automatic undo management enabled, then set this parameter to a unique undo tablespace name for each instance.

Summary of Parameter Use in Real Application Clusters Databases

This section summarizes considerations for using parameters in RAC databases.

CLUSTER_DATABASE Enables a database to be started in cluster mode. Set this parameter to TRUE.

CLUSTER_DATABASE_INSTANCES Sets the number of instances in your RAC environment. A proper setting for this parameter can improve memory use.

CLUSTER_INTERCONNECTS Specifies the cluster interconnect when there is more than one interconnect. Refer to your Oracle platform-specific documentation for the use of this parameter, its syntax, and its behavior.

You typically do not need to set the CLUSTER_INTERCONNECTS parameter. For example, do not set this parameter for the following common configurations:

- If you have only one cluster interconnect.
- If the default cluster interconnect meets the bandwidth requirements of your RAC database, which is typically the case.

Oracle uses information from CLUSTER_INTERCONNECTS to distribute interconnect traffic among the various network interfaces if you specify more than one interconnect with this parameter. Note that the specified configuration inherits any limitations of the listed interconnects and the associated operating system IPC services, such as availability. Consider setting CLUSTER_INTERCONNECTS when a single cluster interconnect cannot meet your bandwidth requirements. You may need to set this parameter in data warehouse environments with high interconnect bandwidth demands from one or more databases as described here.

For example, if you have two databases with high interconnect bandwidth requirements, then you can override the default interconnect provided by your

operating system and nominate a different interconnect for each database using the following syntax in each server parameter file where *ip_n* is an IP address in standard dot-decimal format, for example: 144 . 25 . 16 . 214:

```
Database One: CLUSTER_INTERCONNECTS = ip1  
Database Two: CLUSTER_INTERCONNECTS = ip2
```

If you have one database with high bandwidth demands, then you can nominate multiple interconnects using the following syntax:

```
CLUSTER_INTERCONNECTS = ip1:ip2:...:ipn
```

If you set multiple values for `CLUSTER_INTERCONNECTS` as in the preceding example, then Oracle uses all of the interconnects that you specify. This provides load balancing as long as all of the listed interconnects remain operational.

If there is an operating system error writing to the interconnect that you specify with `CLUSTER_INTERCONNECTS`, then Oracle returns an error even if some other interfaces are available. This is because the communication protocols between Oracle and the interconnect can vary greatly depending on your platform. Refer to your Oracle platform-specific documentation for more information.

DB_NAME If you set a value for `DB_NAME` in instance-specific parameter files, the setting must be identical for all instances.

DISPATCHER Set the `DISPATCHERS` parameter to enable a shared server configuration, that is a server that is configured to allow many user processes to share very few server processes. With shared server configurations, many user processes connect to a dispatcher. The `DISPATCHERS` parameter may contain many attributes.

Oracle recommends that you configure at least the `PROTOCOL` and `LISTENER` attributes. `PROTOCOL` specifies the network protocol for which the dispatcher process generates a listening end point. `LISTENER` specifies an alias name for the Oracle Net Services listeners. Set the alias to a name that is resolved through a naming method such as a `tnsnames.ora` file. The `tnsnames.ora` file contains net service names. This file is needed on clients, nodes, the Enterprise Manager Central Control, and the Oracle Performance Manager node. Refer to *Oracle Net Services Administrator's Guide* for complete information about configuring the `DISPATCHER` parameter and its attributes and for configuring the shared server.

MAX_COMMIT_PROPAGATION_DELAY This is a RAC-specific parameter. Do not alter the default setting for this parameter except under a limited set of circumstances. This parameter specifies the maximum amount of time allowed before the system change number (SCN) held in the SGA of an instance is refreshed by the log writer process (LGWR). It determines whether the local SCN should be refreshed from the SGA when getting the snapshot SCN for a query.

SPFILE When you use an `SPFILE`, all RAC database instances must use the `SPFILE` and the file must be on shared storage.

SESSIONS_PER_USER Each instance maintains its own `SESSIONS_PER_USER` count. If `SESSIONS_PER_USER` is set to 1 for a user, the user can log on to the database more than once as long as each connection is from a different instance.

THREAD If specified, this parameter must have unique values on all instances. The `THREAD` parameter specifies the number of the redo thread to be used by an instance. You can specify any available redo thread number as long as that thread number is enabled and is not used.

Backing Up the Server Parameter File

Oracle recommends that you regularly back up the server parameter file for recovery purposes. Do this using the `CREATE PFILE` statement. For example:

```
CREATE PFILE='?/dbs/initdbname.ora'  
FROM SPFILE='/dev/vx/rdisk/oracle_dg/dbspfile'
```

You can also recover by starting up an instance using a client-side initialization parameter file. Then re-create the server parameter file using the `CREATE SPFILE` statement. You can also use RMAN (Recovery Manager) to create backups of the server parameter file.

See Also:

- *Oracle Database Backup and Recovery Basics* for more information about RMAN
- *Oracle Database SQL Reference* for more information about the `CREATE SPFILE` statement

Administering Storage

This chapter describes storage topics such as Automated Storage Management (ASM) and the Oracle Cluster Registry (OCR). The topics in this chapter are:

- [Overview of Storage in Oracle Real Application Clusters](#)
- [Automatic Storage Management](#)
- [Managing the Oracle Cluster Registry in Real Application Clusters](#)

Overview of Storage in Oracle Real Application Clusters

Storage for RAC databases must be shared. In other words, datafiles must reside on either a cluster file system or on shared raw devices. Additionally, for each instance you must create at least two redo log files that reside on shared storage devices. You must also create one shared device for each instance to store its own tablespace for automatic undo management. If needed, you can use a local file system to store, for example, client-side parameter files (PFILES).

Unless otherwise noted, Oracle storage features such as Automatic Storage Management (ASM), Oracle Managed Files, automatic segment-space management, and so on, function the same in RAC databases as they do in Oracle single-instance databases. Refer to the *Oracle Database Administrator's Guide* for additional information about these storage features.

If your platform does not support a cluster file system or if you do not want to use a cluster file system to store datafiles for RAC, then you must create additional raw devices as described in *Oracle 2 Day DBA* and *Oracle Real Application Clusters Installation and Configuration Guide*. Oracle recommends that you use ASM for datafile storage as described in the next section.

Automatic Storage Management

ASM automatically optimizes storage to maximize performance by rebalancing the storage configuration across the disks that ASM manages. ASM spreads the storage load across all available storage within your cluster database environment for optimal performance. ASM partitions your total disk space into uniformly sized units across all disks in a disk group. ASM can also automatically mirror data to prevent data loss.

To use ASM in RAC, you only need to select ASM as your storage option when you create your database with the Database Configuration Assistant (DBCA). When your database is operating, storage management only requires you to administer disk groups as described in this section. As in single-instance Oracle databases, using ASM in RAC does not require I/O tuning.

Automatic Storage Management Components in RAC

When you use ASM, there is an ASM instance on each node of a RAC database. Each ASM instance has either an SPFILE or PFILE type parameter file. You may need to back up the parameter files and the TNS entries if you are using a non-default Oracle Net Listener.

Modifying Disk Group Configurations for ASM in RAC

When you add disk groups in RAC, you only need to prepare the underlying operating system. ASM automatically rebalances the storage load after you add a disk group or when you delete a disk group.

When you add disk groups, you can also set a redundancy level and identify failure groups which are sets of disks that share a common resource. ASM uses failure groups to store redundant copies of your data to ensure that the loss of one failure group does not result in data loss. The default redundancy level is normal, which mirrors to two failure groups. You can also specify high redundancy for three-way mirroring. You can only use failure groups if you enable mirroring.

In RAC, each ASM instance manages all disk and failure groups for each node. In addition, each ASM instance coordinates disk and failure group information with other nodes in a RAC cluster database. As in single-instance Oracle databases, you can

use the DBCA, SQL*Plus, and the Server Control Utility (SRVCTL) to administer disk groups for ASM in RAC as described in the following sections.

Administering ASM Instances and ASM Disk Groups with Enterprise Manager in RAC

You can add or administer Automatic Storage Management (ASM) disk groups with Enterprise Manager. You can also monitor ASM disk group performance and control disk group availability at the instance level. For example, the additional ASM-related Enterprise Manager tasks that you can perform in RAC are:

- When you add a disk group, the disk group definition includes a checkbox to indicate whether the disk group is automatically mounted to all of the cluster database instances.
- When you examine the default Disk Group Performance page, you can see an instance-level performance details by clicking a performance characteristic such as Write Response Time or I/O Throughput.
- When you mount and dismount ASM disk groups, you can use a checkbox to indicate which instances mount or dismount the ASM Disk Group.

See Also: *Oracle Database Administrator's Guide* and *Oracle 2 Day DBA* for detailed information about using Enterprise Manager

Administering ASM Instances with SRVCTL in RAC

You can use the Server Control Utility (SRVCTL) to add, remove, enable, and disable an ASM instance as described in the following procedures:

Use the following syntax to add configuration information about an existing ASM instance:

```
srvctl add asm -n <node_name> -i <asm_instance_name> -o <oracle_home>
```

Note: For all of the SRVCTL commands in this section for which the -i option is not required, if you do not specify an instance name, then the command applies to all the ASM instances on the node.

Use the following syntax to remove an ASM instance:

```
srvctl remove asm -n <node_name> [-i <asm_instance_name>]
```

Use the following syntax to enable an ASM instance:

```
srvctl enable asm -n <node_name> [-i <asm_instance_name>]
```

Use the following syntax to disable an ASM instance:

```
srvctl disable asm -n <node_name> [-i <asm_instance_name>]
```

You can also use SRVCTL to start, stop, and obtain the status of an ASM instance as in the following examples.

Use the following syntax to start an ASM instance:

```
srvctl start asm -n <node_name> [-i <asm_instance_name>] [-o <start_options>]
```

Use the following syntax to stop an ASM instance:

```
srvctl stop asm -n <node_name> [-i <asm_instance_name>] [-o  
<stop_options>]
```

Use the following syntax to configure an ASM instance:

```
srvctl config asm -n <node_name>
```

Use the following syntax to obtain the status of an ASM instance:

```
srvctl status asm -n <node_name>
```

Datafile Access in Real Application Clusters

All RAC instances must be able to access all datafiles. If a datafile needs to be recovered when the database is opened, then the first RAC instance to start performs the recovery and verifies access to the file. As other instances start up, they also verify their access to the datafiles. Similarly, when you add a tablespace or datafile or bring a tablespace or datafile online, all instances verify access to the file or files.

If you add a datafile onto a disk that other instances cannot access, then verification fails. Verification also fails if instances access different copies of the same datafile. If verification fails for any instance, then diagnose and fix the problem. Then execute the `ALTER SYSTEM CHECK DATAFILES` statement on each instance in your RAC database to verify datafile access.

Redo Log File Storage in Real Application Clusters

Each instance has its own online redo log groups which are called an instance's thread of online redo. Create these online redo log groups and establish group members as described in the *Oracle Database Administrator's Guide*.

Each instance must have at least two groups of online redo log files in its own thread. When the current group fills, an instance begins writing to the next log file group. If your database is in ARCHIVELOG mode, then each instance must save filled log files into its own archive log thread and update the control file with the status of its thread.

Note: `MAXLOGHISTORY` is useful for sites with demanding availability requirements. This option can help you administer recovery, especially when there are many instances and many log files.

Automatic Undo Management in Real Application Clusters

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, all instances can always read all undo blocks for consistent read purposes. Also, any instance can update any undo tablespace during transaction recovery, as long as that undo tablespace is not currently used by another instance for undo generation or transaction recovery.

You assign undo tablespaces in your 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 a RAC database. In other words, all instances of a RAC database must operate in the same undo mode.

See Also: *Oracle Database Administrator's Guide* for detailed information about creating and managing undo tablespaces

Switching Undo Tablespaces for Instances in Real Application Clusters

You can dynamically switch undo tablespace assignments. Redirect the assignment for an instance by executing the `ALTER SYSTEM SET UNDO_TABLESPACE` statement. You might do this during planned maintenance to redirect an instance to use an idle undo tablespace. For example, assume that you have specified instances `db1` and `db2` to access undo tablespaces `undotbs01` and `undotbs02` respectively, and that you have an idle undo tablespace `undotbs03`. To redirect instance `db1` to use undo tablespace `undotbs03`, execute the following SQL statement from instance `db1`:

```
ALTER SYSTEM SET UNDO_TABLESPACE = undotbs3 SID=1;
```

In this example, the previously used undo tablespace `undotbs01` remains assigned to instance `db1` until the instance's last active transaction commits. User transactions proceed while Oracle performs the switching operation as follows:

- The switching process does not wait for all user transactions to commit.
- Oracle places the previous undo tablespace in a pending-offline state if there are active transactions in that tablespace. This means that the pending-offline tablespace may be unavailable for other instances until all transactions against that tablespace are committed.

Refer to the *Oracle Database Administrator's Guide* for more information about the `ALTER SYSTEM SET UNDO_TABLESPACE` statement and information about setting the undo retention period and dropping undo tablespaces.

Managing the Oracle Cluster Registry in Real Application Clusters

This section describes managing the Oracle Cluster Registry (OCR). The OCR contains cluster and database configuration information for RAC and Cluster Ready Services (CRS) such as the cluster node list, and cluster database instance to node mapping, and CRS application resource profiles.

There are two methods of copying OCR content and using the content for recovery. The first method uses automatically generated physical OCR file copies and the second method uses manually created logical OCR export files as described in the following sections:

- [Managing Oracle Cluster Registry Backup Files in Real Application Clusters](#)
- [Managing the Oracle Cluster Registry with OCR Exports](#)

Managing Oracle Cluster Registry Backup Files in Real Application Clusters

One CRS instance in a cluster automatically creates OCR backups every four hours. At any one time, the last three backup copies are always retained. The CRS instance also creates and retains an OCR backup for each *full day* and at the end of each *week*. You cannot customize the backup frequencies or the number of files that Oracle retains. However, because of the importance of OCR information, Oracle recommends that you use the `ocrconfig` tool to make copies of the automatically generated backup files, at least once daily. The `ocrconfig` tool commands are described later in this section.

Use the `ocrconfig` tool with the `showbackup` option to identify the backup files. Then copy each file to a location that is redundant to the location of the original OCR backup file. This ensures that there are at least two copies of each OCR backup file. Oracle also recommends that the OCR location reside on RAID arrays. If possible, use a backup location that is shared by all nodes in the cluster. The default target location

of each OCR backup file is as follows where `<cluster name>` is the name you assigned to the cluster when you installed CRS:

```
<CRS Home>/cdata/<cluster name>
```

You can change this location using the following syntax:

```
ocrconfig -backuploc <directory name>
```

Restoring the Oracle Cluster Registry from Automatically Generated OCR Backups

Before attempting to resolve any configuration-related problems, retry the application that failed. If the problem continues, then use one of the following platform-specific procedures to restore the OCR configuration.

- [Restoring the Oracle Cluster Registry on UNIX-Based Systems](#)
- [Restoring the Oracle Cluster Registry on Windows-Based Systems](#)

Note: You cannot restore your configuration from an automatically created OCR backup file using the `-import` option. You must instead use the `-restore` option.

Restoring the Oracle Cluster Registry on UNIX-Based Systems

1. Stop the following OCR clients: CSSD, CRSD, EVMD, and the EVM logger daemons.
2. Shut down all the nodes in the cluster and restart from one node in single-user mode.
3. Identify the recent backups using the `ocrconfig -showbackup` command.
4. Execute the restore by applying an OCR backup file identified in Step 3 with the `ocrconfig -restore <file name>` command.
5. Restart the CRS, CSS, and EVM daemons and resume operations in cluster mode.

Restoring the Oracle Cluster Registry on Windows-Based Systems

1. Shut down all but one node in your RAC database.
2. Disable the following OCR clients and stop them using the Service Control Panel: OracleClusterVolumeService, OracleCSService, OracleCRService, and the OracleEVMSservice.
3. Identify the recent backups using the `ocrconfig -showbackup` command.
4. Execute the restore by applying an OCR backup file identified in Step 3 with the `ocrconfig -restore <file name>` command.
5. Start all the services that were stopped in step 2. Restart all the instances and resume operations in cluster mode.

Managing the Oracle Cluster Registry with OCR Exports

In addition to using the automatically created OCR backup files, you should also export the OCR contents before and after making significant configuration changes. Do this by using the `ocrconfig -export` command. This enables you to 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 clusterware, restore your configuration using one of the following platform-specific procedures:

- [Importing Oracle Cluster Registry Content on UNIX-Based Systems](#)
- [Importing Oracle Cluster Registry Content on Windows-Based Systems](#)

Note: You cannot import an exported OCR backup file using the `-restore` option. You must instead use the `-import` option.

Importing Oracle Cluster Registry Content on UNIX-Based Systems

1. Shut down all the nodes in the cluster and restart one of them in single-user mode.
2. Import an OCR export file using the `ocrconfig -import` command from any node.
3. Start all the nodes in the cluster in multi-user mode.

To avoid stopping the entire cluster, use this alternate method:

1. Update `inittab` entries on all the nodes and remove the CRS-specific entries. Retain a copy of the original `inittab` file.
2. Run the `/etc/init.d/init.crs stop` command on all the nodes.
3. Import the OCR export file using the `ocrconfig -import` command from any node.
4. Restore the original `inittab` file on all nodes.
5. Run the `/etc/init.d/init.crs start` command on all nodes.
6. Run the `/etc/init q` command.

Importing Oracle Cluster Registry Content on Windows-Based Systems

1. Stop the following OCR clients using the Service Control Panel: OracleClusterVolumeService, OracleCMService, OracleEVMSERVICE, OracleCSService, and the OracleCRService.
2. Import an OCR export file using the `ocrconfig -import` command from one node.
3. Re-start all of the affected services on all nodes.

The ocrconfig Tool Command Syntax and Options

Execute OCR commands with administrative privileges on UNIX-based platforms or as a user with Administrator privileges on Windows-based systems. The OCR command syntax is as follows where `<options>` is a verb in [Table 3-1](#):

```
ocrconfig -<option>
```

Table 3-1 The ocrconfig Command Options

Option	Purpose
<code>-export</code>	To export the contents of the OCR into a target file.
<code>-import</code>	To import OCR contents from a previously <i>exported</i> OCR file.
<code>-restore</code>	To restore the OCR from an automatically created OCR backup file.

Table 3–1 (Cont.) The ocrconfig Command Options

Option	Purpose
-backuploc	To change OCR backup file location. For this entry, use a full path that is accessible by all nodes.
-showbackup	To display the location, timestamp, and the node name of origin for the last three automatically created backup files.
-upgrade	To upgrade the OCR to a later version.
-downgrade	To downgrade the OCR to an earlier version.
-help	To display help for the ocrconfig commands.

For example, to export the OCR contents to a binary file, use the `ocrconfig` command with the following syntax where `file_name` is the file to which you want to export the OCR contents as follows:

```
ocrconfig -export file_name
```

Do not attempt to edit the export file. If OCR clients such as the clusterware are running, then Oracle reports an error.

Implementing the Oracle Hardware Assisted Resilient Data Initiative for the OCR

The Oracle Hardware Assisted Resilient Data (HARD) initiative prevents data corruptions before they occur. With HARD enabled, the OCR writes HARD-compatible blocks. To determine whether the device used by the OCR supports HARD and enable it, review the Oracle HARD white paper at:

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

Upgrading and Downgrading the OCR Configuration in Real Application Clusters

When you install CRS, Oracle automatically executes the `ocrconfig -upgrade` command. To downgrade, follow the downgrade instructions for each component and also downgrade the OCR using the `ocrconfig -downgrade` command.

HARD-Compatible OCR Blocks in Oracle9i

In Oracle9i, the OCR did not write HARD-compatible blocks. If the device used by OCR is enabled for HARD, then use the method described in the HARD white paper to *disable* HARD for the OCR device before downgrading your OCR. If you do not disable HARD, then the downgrade operation fails.

Importing and Exporting Cluster Database Configuration Information with SRVCONFIG

Use `SRVCONFIG` to import and export SRVM cluster database configuration information to and from the OCR. For example, the following syntax **exports** the configuration information to a text file:

```
srvconfig -exp file_name.txt
```

The following syntax **imports** the configuration information from a text file into the OCR:

```
srvconfig -imp file_name.txt
```

Administering Services

This chapter describes how to administer high availability features such as services within Oracle Real Application Clusters (RAC) databases. The topics in this chapter are:

- [Services and High Availability in Real Application Clusters](#)
- [Adding and Modifying Services](#)
- [Automatic Restarts after Failures](#)
- [Administering Services with Enterprise Manager and SRVCTL](#)

See Also: *Oracle Real Application Clusters Installation and Configuration Guide* for information about installing and configuring RAC

Services and High Availability in Real Application Clusters

Services are groups or classifications of applications that comprise business components that extend and respond to application workloads. Examples of services are Accounts Payable (AP), Customer Relationship Management (CRM), and so on. Services in RAC enable continuous, uninterrupted database operations to support multiple services on multiple instances.

Services enable RAC to integrate cluster database resources into a single system image to optimize cluster manageability. This simplifies system deployment, testing, and disaster recovery and administrative overhead. With services, users connect to a database without regard for which instance executes the SQL application service.

You assign services to run on one or more instances, and alternate instances can serve as backup instances in case the primary instance fails. If a primary instance fails, then Oracle moves the service from the failed instance to a surviving alternate instance.

Services enable you to model and deploy both planned and unplanned operations for all types of high availability or disaster recovery scenarios. During outages, RAC automatically restarts key components. Components that are eligible for automatic restart include instances, Oracle Net Services listeners, and the database as well as several database subcomponents.

To respond to changes in your cluster database, RAC uses events for rapid notification of state changes. After failures, event notifications initiate recovery processing to eliminate network timeouts and to provide end-to-end recovery of critical applications.

Because key resources are linked by RAC high availability events, resources respond to these events regardless of where the events occur in your environment. You can also define events to extend notification to middle-tier applications to simplify monitoring.

You can also modify your cluster configuration and still maintain service availability due to the resiliency of client-to-database connections. This is because RAC continually propagates both service and cluster configuration information throughout your cluster database.

With connect-time and run-time routing, the number of instances that offer a service is transparent to applications. Because the Oracle Resource Manager is integrated with RAC, services also improve the throughput for applications running in RAC environments.

Adding and Modifying Services

Use the Database Configuration Assistant (DBCA) or Server Control (SRVCTL) to create services. Then, depending on the task you want to perform, use the DBCA or Enterprise Manager to administer them. You can measure resource usage at the service level and control services availability by setting performance thresholds.

When you create services in RAC, you can assign the services to instances for preferred (normal) and available (recovery) processing. You can identify other instances that are available to support the service if service levels change or for planned outages. Additionally, you can use the Oracle Resource Manager to create consumer groups that control the priority for the service. When a service's preferred instance becomes unavailable, Oracle re-connects users from the unavailable instance to an available instance.

If you use the Database Upgrade Assistant (DBUA) to upgrade a Primary/Secondary configuration from an earlier release of Oracle Database, then the DBUA will create

one service and assign it to one instance as a preferred instance and to the other instance as an available instance.

See Also: *PL/SQL Packages and Types Reference* for more information about the `DBMS_SERVICE` PL/SQL and `DBMS_MONITOR` packages and for more information about setting thresholds

Using the DBCA to Add and Modify Services

The DBCA Service Management feature enables you to manage service assignments and service preferences for instances and to configure Transparent Application Failover (TAF) policies. You can execute these procedures while your RAC database is running. Even if your instances or the RAC database is not running, you can still use the DBCA to configure services, but the services will not start. To add or modify services using the DBCA Services Management feature:

1. On the DBCA Welcome page, select the Oracle Real Application Clusters option and click Next.
2. On the DBCA Operations page, select Services Management and click Next.
3. On the DBCA List of Databases page, select the cluster database for which you want to configure services and click Next. If the database that you selected already has services assigned to it, then the DBCA displays these on this page.
4. Click Add or Modify.
5. To add a service, enter the name of the service. Note that service names with the prefix `SYS$` are reserved for use by Oracle internal processes. To modify a service, edit the service and configure the service's instance preferences and TAF policies. Assign the service to instances for preferred (normal) and available (recovery) processing. The DBCA records your changes when you select another service or proceed to another page.

Note: Entries you make in the Add a Service dialog are appended to the `SERVICE_NAMES` parameter entry which has a 4KB limit. Therefore, the total length of the names of all services assigned to an instance cannot exceed 4KB.

6. To remove a service, select the service and click Remove.
7. Click Finish and the DBCA displays the Summary page. Click OK and the DBCA displays a progress dialog while it configures your services.

When you click Finish, the DBCA configures the Cluster Ready Services (CRS) resources for the services that you added, modified, or removed. The DBCA also configures the net service entries for these services and starts them. When you use the DBCA to remove services, the DBCA stops the service, removes the CRS resource for the service, and removes the net service entries.

Automatic Restarts after Failures

Some administrative tasks require that you prevent components from automatically re-starting. For example, to take a node and all of its instances and services offline for maintenance purposes, first disable the instance and its services using either Enterprise Manager or `SRVCTL`.

If a failure occurs and adversely affects a RAC database instance, then RAC moves any services on that instance to another available instance. Then CRS attempts to restart the failed software when the node comes back up.

Note: If you stop an instance to perform administration, you *must* disable the instance with SRVCTL. Otherwise, if the node fails and then restarts, then CRS attempts to restart the instance during your administrative operation.

Administering Services with Enterprise Manager and SRVCTL

When you create a service with SRVCTL, you must start it with a separate SRVCTL command. However, you may later need to manually stop or restart the service. In addition, you may need to disable the service to prevent automatic restarts. You may also need to manually relocate the service or obtain status information about the service. Refer to the following sections for examples of how to start and stop services.

- [Administering Services in Real Application Clusters with Enterprise Manager](#)
[Administering Services in Real Application Clusters with SRVCTL](#)

Administering Services in Real Application Clusters with Enterprise Manager

Use the Enterprise Manager service administration page to:

- Enable, disable, start, stop, and relocate cluster database services
- View the services that are available on the instances of a cluster database
- View the lists of running, preferred, and available instances that were configured during service definition
- View the TAF policy set for each service.

To access the Cluster Managed Database Services page, from the Enterprise Manager Home page, under View select Cluster Databases and click Go. This takes you to a cluster database's home page. Click the Administration tab and on the Cluster Database Administration page, under the High Availability options list in the right, click Cluster Managed Database Services.

The Cluster Managed Database Services page displays services that are available on cluster database instances. Also use this page to manually load balance services. On the Services Detail page, select the service that you want to administer using the radio buttons on the left-hand side of the panel and select an operation by clicking the command buttons on the right-hand side of the panel, such as ENABLE, DISABLE, and so on, and click OK. When you click OK, Enterprise Manager performs the operation that you selected on the service. Refer to the online help for more information about this page.

You can only administer a service with Enterprise Manager after creating the service with the DBCA or SRVCTL. In addition, to use TAF, you must have previously configured the service for TAF; you cannot use Enterprise Manager to create services or to set service TAF policies. To access the Services Details page:

- To start or stop a service, access the service by way of the Cluster Database page Administration tab and click Start or Stop.

See Also:

- ["Using the DBCA to Add and Modify Services"](#) on page 4-3 for information about using the DBCA to create services
- ["srvctl add"](#) on page B-5 for more information about using SRVCTL to add services
- *PL/SQL Packages and Types Reference* for more information about the DBMS_SERVICE PL/SQL package

Administering Services in Real Application Clusters with SRVCTL

The following sections explain how to use SRVCTL to administer services.

Starting and Stopping Services with SRVCTL

Enter the following SRVCTL syntax from the command line:

```
srvctl start service -d <name> [-s <service_name_list>] [-i <inst_name>]
[-o <start_options>] [-c <connect_str> | -q]
```

```
srvctl stop service -d <name> -s <service_name_list> [-i <inst_name>]
[-o <start_options>] [-c <connect_str> | -q]
```

Enabling and Disabling Services with SRVCTL

By default, all services are enabled for automatic restart. However, for some administrative activities such as hardware repairs and planned upgrades, you must disable a service to prevent automatic restart. Use the following SRVCTL syntax from the command line to enable and disable services:

```
srvctl enable service -d <db_name> -s <service_name1,service_name2,...>
[-i <inst_name1,inst_name2,...>]
```

```
srvctl disable service -d <db_name> -s <service_name1,service_name2,...>
[-i <inst_name1,inst_name2,...>]
```

Relocating Services with SRVCTL

Use the following SRVCTL syntax from the command line to relocate a service, for example, from instance crm1 to instance crm3:

```
srvctl relocate service -d crm -s crm -i crm1 -t crm3
```

```
srvctl status service -d crm -s crm -v
```

You can perform other administrative tasks with SRVCTL as described in [Appendix B, "Server Control \(SRVCTL\) Reference"](#).

Obtaining the Statuses of Services with SRVCTL

Use the following SRVCTL syntax from the command line to obtain the status of, for example, the CRS service that is running on crm:

```
srvctl status service -d crm -s crm
```

Adding and Deleting Nodes and Instances

This chapter describes how to add and delete nodes and instances in Oracle Real Application Clusters (RAC) databases. The topics in this chapter are:

- [Overview of Node Addition Procedures](#)
- [Step 1: Connecting New Nodes to the Cluster](#)
- [Step 2: Extending Clusterware and Oracle Software to New Nodes](#)
- [Step 3: Preparing Storage for RAC on New Nodes](#)
- [Step 4: Adding Nodes at the Oracle RAC Database Layer](#)
- [Step 5: Adding Database Instances to New Nodes](#)
- [Adding Nodes that Already Have Clusterware and Oracle Software to a Cluster](#)
- [Deleting Instances from Real Application Clusters Databases](#)
- [Deleting Nodes from Oracle Clusters on UNIX-Based Systems](#)
- [Deleting Nodes from Oracle Clusters on Windows-Based Platforms](#)

Overview of Node Addition Procedures

This section explains how to add nodes to clusters. Do this by setting up the new nodes to be part of your cluster at the network level. Then extend the Cluster Ready Services (CRS) home from an existing CRS home to the new nodes and then extend the Oracle database software with RAC components to the new nodes. Finally, make the new nodes members of the existing cluster database.

Note: You can add nodes on some UNIX-based platforms without stopping existing nodes if your clusterware supports this. Refer to your vendor-specific clusterware documentation for more information.

If the nodes that you are adding to your cluster do not have clusterware or Oracle software, then you must complete the following five steps. The procedures in these steps assume that you already have an operative UNIX-based or Windows-based RAC environment. The details of these steps appear in the following sections.

- [Step 1: Connecting New Nodes to the Cluster](#)
- [Step 2: Extending Clusterware and Oracle Software to New Nodes](#)
- [Step 3: Preparing Storage for RAC on New Nodes](#)
- [Step 4: Adding Nodes at the Oracle RAC Database Layer](#)
- [Step 5: Adding Database Instances to New Nodes](#)

To add a node to your cluster when the node is already configured with clusterware and Oracle software, follow the procedure described in "[Adding Nodes that Already Have Clusterware and Oracle Software to a Cluster](#)" on page 5-12.

See Also: *The Oracle Real Application Clusters Installation and Configuration Guide* for procedures about using the Database Configuration Assistant (DBCA) to create and delete RAC databases

Step 1: Connecting New Nodes to the Cluster

Complete the following procedures to connect the new nodes to the cluster and to prepare them to support your cluster database:

- [Making Physical Connections](#)
- [Installing Operating System](#)
- [Creating Oracle Users](#)
- [Checking the Installation](#)

Making Physical Connections

Connect the new nodes' hardware to the network infrastructure of your cluster. This includes establishing electrical connections, configuring network interconnects, configuring shared disk subsystem connections, and so on. Refer to your hardware vendor documentation for details about this step.

Installing Operating System

Install a cloned image of the operating system that matches the operating system on the other nodes in your cluster. This includes installing required service patches and drivers. Refer to your hardware vendor documentation for details about this process.

Creating Oracle Users

As root user on UNIX-based systems, create the Oracle users and groups using the same user ID and group ID as on the existing nodes. On Windows-based systems, perform the installation as an Administrator.

Checking the Installation

To verify that your installation is configured correctly, perform the following steps:

1. Ensure that the new nodes can access the private interconnect. This interconnect must be properly configured before you can complete the procedures in ["Step 2: Extending Clusterware and Oracle Software to New Nodes"](#) on page 5-3.
2. If you are not using a cluster file system, then determine the location on which your cluster software was installed on the existing nodes. Make sure that you have at least 250MB of free space on the same location on each of the new nodes to install the Cluster Ready Services software. In addition, ensure you have enough free space on each new node to install the Oracle binaries.
3. Ensure that user equivalence is established on the new nodes.
4. Execute the following platform-specific procedures:

- On UNIX-based systems:

Verify user equivalence to and from an existing node to the new nodes using `rsh` or `ssh`.

- On Windows-based systems:

Make sure that you can execute the following command from *each* of the existing nodes of your cluster where the `host_name` is the public network name of the new node:

```
NET USE \\host_name\C$
```

You have the required administrative privileges on each node if the operating system responds with:

```
Command completed successfully.
```

After completing the procedures in this section, your new nodes are connected to the cluster and configured with the required software to make them visible to the clusterware. Configure the new nodes as members of the cluster by extending the cluster software to the new nodes as described in Step 2.

Step 2: Extending Clusterware and Oracle Software to New Nodes

The following topics describe how to add new nodes to the clusterware and to the Oracle database software layers using the OUI:

- [Adding Nodes at the Vendor Clusterware Layer \(UNIX only\)](#)
- [Adding Nodes at the Oracle Clusterware Layer \(UNIX and Windows\)](#)

Adding Nodes at the Vendor Clusterware Layer (UNIX only)

If you are using a Windows-based system, then skip this section and proceed to the next section. For UNIX-based systems, add the new nodes at the clusterware layer according to the vendor clusterware documentation. For systems using shared storage for the CRS home, ensure that the existing clusterware is accessible by the new nodes. Also ensure that the new nodes can be brought online as part of the existing cluster. Proceed to the next section to add the nodes at the clusterware layer.

Adding Nodes at the Oracle Clusterware Layer (UNIX and Windows)

On all platforms, complete the following steps. The OUI requires access to the private interconnect that you checked as part of the installation validation in Step 1.

1. On one of the *existing* nodes go to the <CRS home>/OUI/bin directory on UNIX-based systems or to the <CRS home>\oui\bin directory on Windows-based systems. On UNIX run the `addNode.sh` script and on Windows run the `addNode.bat` script to start the OUI.
2. The OUI runs in the `add node` mode and the OUI Welcome page appears. Click Next and the Specify Cluster Nodes for Node Addition page appears.
3. The upper table on the Specify Cluster Nodes for Node Addition page shows the existing nodes associated with the CRS home from which you launched the OUI. Use the lower table to enter the public and private node names of the new nodes.
4. If you are using vendor clusterware, then the public node names automatically appear in the lower table. Click Next and the OUI verifies connectivity on the existing nodes and on the new nodes. The verifications that the OUI performs include determining whether:
 - The nodes are up
 - The nodes are accessible by way of the network
 - The user has write permission to create the CRS home on the new nodes
 - The user has write permission to the OUI inventory in the `oraInventory` directory on UNIX or `Inventory` directory on Windows
5. If the OUI detects that the new nodes do not have an inventory location, then:
 - On UNIX platforms the OUI displays a dialog asking you to run the `oraInstRoot.sh` script on the new nodes
 - On Windows platforms the OUI automatically updates the inventory location in the Registry key

If any verifications fail, then the OUI re-displays the Specify Cluster Nodes for Node Addition page with a Status column in both tables indicating errors. Correct the errors or deselect the nodes that have errors and proceed. However, you cannot deselect existing nodes; you must correct problems on nodes that are already part of your CRS cluster before you can proceed with node addition. If all the checks succeed, then the OUI displays the Node Addition Summary page.

Note: Oracle strongly recommends that you install CRS on *every* node in the cluster that has vendor clusterware installed.

6. The Node Addition Summary page displays the following information showing the products that are installed in the CRS home that you are extending to the new nodes:

- The source for the add node process, which in this case is the CRS home
- The private node names that you entered for the new nodes
- The new nodes that you entered
- The required and available space on the new nodes
- The installed products listing the products that are already installed on the existing CRS home

Click Next and the OUI displays the Cluster Node Addition Progress page.

7. The Cluster Node Addition Progress page shows the status of the cluster node addition process. The table on this page has two columns showing the phase of the node addition process and the phase's status according to the following platform-specific content:

On UNIX-based systems this page shows the following four OUI phases:

- Instantiate Root Scripts—Instantiates `rootaddnode.sh` with the public and private node names that you entered on the Cluster Node Addition page.
- Copy the CRS Home to the New Nodes—Copies the CRS home to the new nodes unless the CRS home is on a cluster file system.
- Run `rootaddnode.sh` and `root.sh`—Displays a dialog prompting you to run `rootaddnode.sh` on the local node from which you are running the OUI. Then you are prompted to run `root.sh` on the new nodes.
- Save Cluster Inventory—Updates the node list associated with the CRS home and its inventory.

On Windows-based systems, this page shows the following three OUI phases:

- Copy CRS Home to New Nodes—Copies the CRS home to the new nodes unless the CRS home is on the Oracle Cluster File System.
- Performs Oracle Home Setup—Updates the Registry entries for the new nodes, creates the services, and creates folder entries.
- Save Cluster Inventory—Updates the node list associated with the CRS home and its inventory.

For all platforms, the Cluster Node Addition Progress page's Status column displays `In Progress` while the phase is in progress, `Suspended` when the phase is pending execution, and `Succeeded` after the phase completes. On completion, click Exit to end the OUI session. After the OUI displays the End of Node Addition page, click Exit to end the OUI session.

8. On Windows-based systems, execute the following command to identify the node names and node numbers that are currently in use:

```
<CRS home>\bin\olsnodes- n
```

Execute the `crssetup.exe` command using the next available node names and node numbers to add CRS information for the new nodes. Use the following syntax for `crssetup.exe` where `I` is the first new node number, `<nodeI>` through `<nodeI+n>` is a list of the nodes that you are adding, `<nodeI-number>` through `<nodeI+n-number>` represent the node numbers assigned to the new

nodes, and `<pnI>` through `<pnI+1>` is the list of private networks for the new nodes:

```
<CRS home>\bin\crssetup.exe
-nn
<nodeI>,<nodeI-number>,<nodeI+1>,<nodeI+1-number>,...<nodeI+n>,<nodeI+n-number>
-pn
<pnI>,<nodeI-number>,<pnI+1>,<nodeI+1-number>,...<pnI+n>,<nodeI+n-number>
```

These are the private network names or IP addresses that you entered in Step 3 of this procedure in the Specify Cluster Nodes for Node Addition page. For example:

```
crssetup.exe -nn node3,3,node4,4 -pn node3_pvt,3,node4_pvt,4
```

9. On all platforms, execute the `racgons` utility from the `bin` subdirectory of the CRS home to configure the Oracle Notification Services (ONS) port number as follows:

```
racgons <nodeI>:4948 <nodeI+1>:4948 ... <nodeI+n>:4948
```

After you have completed the procedures in this section for adding nodes at the Oracle clusterware layer, you have successfully extended the CRS home from your existing CRS home to the new nodes. Proceed to Step 3 to prepare storage for RAC on the new nodes.

Step 3: Preparing Storage for RAC on New Nodes

To extend an existing RAC database to your new nodes, configure storage for the new nodes so that the storage is the same as on the existing nodes. For example, the Oracle Cluster Registry (OCR) and the voting disk must be accessible by the new nodes using the same path as the other nodes use. In addition, the OCR and voting disk devices must have the same permissions as on the existing nodes. Prepare the same type of storage on the new nodes as you are using on the other nodes in the RAC environment that you want to extend as follows:

- Automatic Storage Management (ASM)

If you are using ASM, then make sure that the new nodes can access the ASM disks with the same permissions as the existing nodes.

- Oracle Cluster File System (OCFS)

If you are using Oracle Cluster File Systems, then make sure that the new nodes can access the cluster file systems in the same way that the other nodes access them.

- Vendor Cluster File Systems

If your cluster database uses vendor cluster file systems, then configure the new nodes to use the vendor cluster file systems. Refer to the vendor clusterware documentation for the pre-installation steps for your platform.

- Raw Device Storage

If your cluster database uses raw devices, then prepare the new raw devices by following the procedures described in the following section.

See Also: *The Oracle Real Application Clusters Installation and Configuration Guide* for more information about Oracle Cluster File System

Raw Device Storage Preparation for New Nodes

To prepare raw device storage on the new nodes, you need at least two new disk partitions to accommodate the redo logs for each new instance. Make these disk partitions the same size as the redo log partitions that you configured for the existing nodes' instances. Also create an additional logical partition for the undo tablespace for automatic undo management.

On applicable operating systems, you can create symbolic links to your raw devices. Optionally, on all platforms you can create a raw device mapping file and set the `DBCA_RAW_CONFIG` environment variable so that it points to the raw device mapping file.

See Also: The *Oracle Real Application Clusters Installation and Configuration Guide* for more information about configuring raw partitions and using raw device mapping files

Configure Raw Storage on UNIX-Based Systems

Use your vendor-supplied tools to configure the required raw storage.

Configure Raw Partitions on Windows-Based Systems

Perform the following steps from one of the existing nodes of the cluster:

1. Create or identify an extended partition.
2. Click inside an unallocated part of the extended partition.
3. Choose Create from the Partition menu. A dialog box appears in which you should enter the size of the partition. Ensure you use the same sizes as those you used on your existing nodes.
4. Click the newly created partition and select Assign Drive Letter from the Tool menu.
5. Select Don't Assign Drive Letter, and click OK.
6. Repeat steps 2 through 5 for the second and any additional partitions.
7. Select Commit Changes Now from the Partition menu to save the new partition information.
8. Create symbolic links so that the existing nodes and new nodes can recognize the new partitions you just created and the new nodes can recognize the pre-existing symbolic links to logical drives by following these steps:
 - a. Start the Object Link Manager (OLM) by entering the following command from the `<CRS home>\bin` directory on one of the existing nodes:


```
GUIOracleOBJManager
```
 - b. The OLM starts and automatically detects the symbolic links to the logical drives and displays them in OLM's graphical user interface.
 - c. Recall the disk and partition numbers for the partitions that you created in the previous section. Look for the disk and partition numbers in the OLM page and perform the following tasks:
 - Right-click next to the box under the New Link column and enter the link name for the first partition.
 - Repeat the previous step for the second and any additional partitions.

For example, if your RAC database name is *db* and it consists of two instances running on two nodes and you are adding a third instance on the third node, then your link names for your redo logs should be *db_redo3_1*, *db_redo3_2*, and so on.

- To enable automatic undo management for a new node's instance, enter the link name for the logical partition for the undo tablespace that you created in the previous section. For example, if your RAC database name is *db* and if it has two instances running on two nodes and you are adding a third instance on a third node, then your link name for the undo tablespace should be *db_undo3*.
- Select Commit from the Options menu. This creates the new links on the current node.
- Select Sync Nodes from the Options menu. This makes the new links visible to all of the nodes in the cluster, including the new nodes.
- Select Exit from the Options menu to exit the Object Link Manager.

After completing the procedures in this section, you have configured your cluster storage so that the new nodes can access the Oracle software. Additionally, the existing nodes can access the new nodes and instances. Use the OUI as described in the procedures in Step 4 to configure the new nodes at the RAC database layer.

Step 4: Adding Nodes at the Oracle RAC Database Layer

To add nodes at the Oracle RAC database later, run the OUI in add node mode to configure your new nodes. If you have multiple Oracle homes, then perform the following steps for each Oracle home that you want to include on the new nodes:

1. On an existing node from the `$ORACLE_HOME/OUI/bin` directory on UNIX-based systems, run the `addNode.sh` script. From the `%ORACLE_HOME%\oui\bin` on Windows-based systems, run the `addNode.bat` script. This starts the OUI in the add node mode and displays the OUI Welcome page. Click Next on the Welcome page and the OUI displays the Specify Cluster Nodes for Node Addition page.
2. The Specify Cluster Nodes for Node Addition page has a table showing the existing nodes associated with the Oracle home from which you launched the OUI. A node selection table appears on the bottom of this page showing the nodes that are available for addition. Select the nodes that you want to add and click Next.
3. The OUI verifies connectivity and performs availability checks on both the existing nodes and on the nodes that you want to add. Some of checks performed determine whether:
 - The nodes are up
 - The nodes are accessible by way of the network
 - The user has write permission to create the Oracle home on the new nodes
 - The user has write permission to the OUI inventory in the `oraInventory` directory on UNIX or the `Inventory` directory on Windows on the existing nodes and on the new nodes
4. If the new nodes do not have an inventory set up, then on UNIX-based systems the OUI displays a dialog asking you to run the `oraInstRoot.sh` script on the new nodes. On Windows-based systems the OUI automatically updates the Registry entries for the inventory location. If any of the other checks fail, then fix the

problem and proceed or deselect the node that has the error and proceed. You cannot deselect existing nodes; you must correct problems on the existing nodes before proceeding with node addition. If all of the checks succeed then the OUI displays the Node Addition Summary page.

5. The Node Addition Summary page has the following information about the products that are installed in the Oracle home that you are going to extend to the new nodes:
 - The source for the add node process, which in this case is the Oracle home
 - The existing nodes and new nodes
 - The new nodes that you selected
 - The required and available space on the new nodes
 - The installed products listing all the products that are already installed in the existing Oracle home

Click Finish and the OUI displays the Cluster Node Addition Progress page.

6. The Cluster Node Addition Progress page shows the status of the cluster node addition process. The table on this page has two columns showing the phase of the node addition process and the phase's status according to the following platform-specific content:

On UNIX-based systems, the Cluster Node Addition Progress page shows the following four OUI phases:

- Instantiate Root Scripts—Instantiates the `root.sh` script in the Oracle home by copying it from the local node
- Copy the Oracle Home to the New Nodes—Copies the entire Oracle home from the local node to the new nodes unless the Oracle home is on a cluster file system
- Run `root.sh`—Displays the dialog prompting you to run `root.sh` on the new nodes
- Save Cluster Inventory—Updates the node list associated with the Oracle home and its inventory

On Windows-based systems, the Cluster Node Addition Progress shows the following three OUI phases:

- Copy the Oracle Home To New Nodes—Copies the entire Oracle home to the new nodes unless the Oracle home is on a cluster file system
- Performs Oracle Home Setup—Updates the Registry entries for the new nodes, creates the services, and creates folder entries
- Save Cluster Inventory—Updates the node list associated with the Oracle home and its inventory

For all platforms, the Cluster Node Addition Progress page's Status column displays *Succeeded* if the phase completes, *In Progress* if the phase is in progress, and *Suspended* when the phase is pending execution. After the OUI displays the End of Node Addition page, click Exit to end the OUI session.

7. On UNIX-based systems only, run the `root.sh` script.
8. Execute the `vipca` utility from the `bin` subdirectory of the Oracle home using the `-nodelist` option with the following syntax identifying the complete set of

nodes that are now part of your RAC database beginning with Node1 and ending with <NodeN>:

```
vipca -nodelist Node1,Node2,Node3,...<NodeN>
```

9. If the private interconnect interface names on the new nodes are not the same as the interconnect names that are on the existing nodes, then configure the private interconnect for the new nodes. Do this by executing the `oifcfg` utility with the `setif` option from the `bin` directory of the Oracle home using the following syntax where <subnet> is the subnet for the private interconnect of the RAC databases to which you are adding nodes:

```
oifcfg setif <interface-name>/<subnet>:<cluster_interconnect|public>
```

For example:

```
oifcfg setif hme3/172.168.16.0:cluster_interconnect
```

After completing the procedures in the previous section, you have defined the new nodes at the cluster database layer. You can now add database instances to the new nodes as described in Step 5.

Step 5: Adding Database Instances to New Nodes

Execute the following procedures on each new node to add instances:

1. Start the Database Configuration Assistant (DBCA) by entering `dbca` at the system prompt from the `bin` directory in the `$ORACLE_HOME` on UNIX. On Windows-based systems, choose Start > Programs > Oracle - HOME_NAME > Configuration and Migration Tools > Database Configuration Assistant.

The DBCA displays the Welcome page for RAC. Click Help on any DBCA page for additional information.
2. Select Real Application Clusters database, click Next, and the DBCA displays the Operations page.
3. Select Instance Management, click Next, and the DBCA displays the Instance Management page.
4. Select Add Instance and click Next. The DBCA displays the List of Cluster Databases page that shows the databases and their current status, such as `ACTIVE`, or `INACTIVE`.
5. From the List of Cluster Databases page, select the active RAC database to which you want to add an instance. If your user ID is not operating-system authenticated, then the DBCA prompts you for a user ID and password for a database user that has SYSDBA privileges. If the DBCA prompts you, then enter a valid user ID and password and click Next. The DBCA displays the List of Cluster Database Instances page showing the names of the existing instances for the RAC database that you selected.
6. Click Next to add a new instance and the DBCA displays the Adding an Instance page.
7. On the Adding an Instance page, enter the instance name in the field at the top of this page if the instance name that the DBCA provides does not match your existing instance name sequence. Then select the new node name from the list, click Next, and the DBCA displays the Services Page.

8. Enter the services information for the new node's instance, click Next, and the DBCA displays the Instance Storage page.
9. If you are using raw devices or raw partitions, then on the Instance Storage page select the Tablespaces folder and expand it. Then select the undo tablespace storage object and a dialog appears on the right-hand side. Change the default datafile name to the raw device name for the tablespace.
10. If you are using raw devices or raw partitions or if you want to change the default redo log group file name, then on the Instance Storage page select and expand the Redo Log Groups folder. For each redo log group number that you select, the DBCA displays another dialog box.
 - For UNIX-based systems, enter the raw device name that you created in the section "[Configure Raw Storage on UNIX-Based Systems](#)" on page 5-7 in the File Name field.
 - On Windows-based systems, enter the symbolic link name that you created in the section, "[Configure Raw Partitions on Windows-Based Systems](#)" on page 5-7.
11. If you are using a cluster file system, then click Finish on the Instance Storage page. If you are using raw devices on UNIX-based systems or disk partitions on Windows-based systems, then repeat step 10 for all of the other redo log groups, click Finish, and the DBCA displays a Summary dialog.
12. Review the information on the Summary dialog and click OK. Or click Cancel to end the instance addition operation. The DBCA displays a progress dialog showing the DBCA performing the instance addition operation. When the DBCA completes the instance addition operation, the DBCA displays a dialog asking whether you want to perform another operation.
13. Click No and exit the DBCA, or click Yes to perform another operation. If you click Yes, then the DBCA displays the Operations page.

After you have completed the procedures in this section, the DBCA has successfully added the new instance to the new node and completed the following steps:

- Created and started an ASM instance on each new node if the existing instances were using ASM
- Created a new database instance on each new node
- For Windows-based systems, created and started the required services
- Created and configured high availability components
- Configured and started node applications for the GSD, Oracle Net Services listener, and Enterprise Manager agent
- Created the Oracle Net configuration
- Started the new instance
- Created and started services if you entered services information on the Services Configuration page

After adding the instances to the new nodes using the steps described in this section, perform any needed service configuration procedures as described in [Chapter 4, "Administering Services"](#).

Adding Nodes that Already Have Clusterware and Oracle Software to a Cluster

To add nodes to a cluster that already have clusterware and Oracle software installed on them, you must configure the new nodes with the Oracle software that is on the existing nodes of the cluster. To do this, you must run two versions of an OUI process: one for the clusterware and one for the database layer as described in the following procedures:

1. Add new nodes at the Oracle clusterware layer by running the OUI from the CRS Home on an existing node according to the following platform-specific procedures:
 - On UNIX execute the following command:

```
<CRS home>/oui/bin/addNode.sh -noCopy
```
 - On Windows execute the following command:

```
<CRS home>\oui\bin\addNode.bat -noCopy
```
2. Add new nodes at the Oracle software layer by running the OUI from the Oracle Home as follows:
 - On UNIX, execute:

```
$ORACLE_HOME/oui/bin/addNode.sh -noCopy
```


On Windows, execute

```
%ORACLE_HOME%\oui\bin\addNode.bat -noCopy
```

In the `-noCopy` mode, the OUI performs all add node operations except for the copying of software to the new nodes.

Deleting Instances from Real Application Clusters Databases

The procedures in this section explain how to use the DBCA to delete an instance from a RAC database. To delete an instance:

1. Start the DBCA on a node *other than* the node that hosts the instance that you want to delete. On the DBCA Welcome page select Oracle Real Application Clusters Database, click Next, and the DBCA displays the Operations page.
2. On the DBCA Operations page, select Instance Management, click Next, and the DBCA displays the Instance Management page.
3. On the Instance Management page, Select Delete Instance, click Next, and the DBCA displays the List of Cluster Databases page.
4. Select a RAC database from which to delete an instance. If your user ID is not operating-system authenticated, then the DBCA also prompts you for a user ID and password for a database user that has SYSDBA privileges. If the DBCA prompts you for this, then enter a valid user ID and password. Click Next and the DBCA displays the List of Cluster Database Instances page. The List of Cluster Database Instances page shows the instances associated with the RAC database that you selected and the status of each instance.
5. Select a remote instance to delete and click Finish.

6. If you have services assigned to this instance, then the DBCA Services Management page appears. Use this feature to reassign services from this instance to other instances in the cluster database.
7. Review the information about the instance deletion operation on the Summary page and click OK. Otherwise, click Cancel to cancel the instance deletion operation. If you click OK, then the DBCA displays a Confirmation dialog.
8. Click OK on the Confirmation dialog to proceed with the instance deletion operation and the DBCA displays a progress dialog showing that the DBCA is performing the instance deletion operation. During this operation, the DBCA removes the instance and the instance's Oracle Net configuration. When the DBCA completes this operation, the DBCA displays a dialog asking whether you want to perform another operation.
9. Click No and exit the DBCA or click Yes to perform another operation. If you click Yes, then the DBCA displays the Operations page.

At this point, you have accomplished the following:

- De-registered the selected instance from its associated Oracle Net Services listeners
- Deleted the selected database instance from the instance's configured node
- For Windows-based systems, deleted the selected instance's services
- Removed the Oracle Net configuration
- Deleted the Oracle Flexible Architecture directory structure from the instance's configured node.

Deleting Nodes from Oracle Clusters on UNIX-Based Systems

Use the following procedures to delete nodes from Oracle clusters on UNIX-based systems:

1. If there are instances on the node that you want to delete, then execute the procedures in the section titled [Deleting Instances from Real Application Clusters Databases](#) on page 5-12 before executing these procedures. If you are deleting more than one node, then delete the instances from all the nodes that you are going to delete.
2. After you have deleted the instances from the node that you want to delete, run the command `$ORACLE_HOME/install/rootdeletenode.sh` on the node you are deleting. This deletes the Oracle Cluster Ready Services (CRS) node applications. If you are deleting more than one node from your cluster, then execute the command `$ORACLE_HOME/install/rootdeletenode.sh node1,node2,...,<nodeN>` where `node1` through `<nodeN>` is a comma-separated list of nodes that you want to delete.
3. On the same node that you are deleting, execute the command `runInstaller -updateNodeList ORACLE_HOME=<Home location> CLUSTER_NODES=node1,node2,...<nodeN>` where `node1` through `<nodeN>` is a comma-separated list of nodes that are remaining in the cluster. This list must exclude the nodes that you are deleting.
4. If you are not using a cluster file system for the Oracle home, then on the node that you are deleting, remove the Oracle database software by executing the `rm` command. *Make sure that you are in the correct Oracle home of the node that you are deleting* when you execute the `rm` command. Execute this command on all the nodes that you are deleting.

5. On the node that you are deleting, run the command `<CRS Home>/install/rootdelete.sh` to disable the CRS applications that are on the node. If the `ocr.loc` file is on a shared file system, then execute the command `<CRS home>/install/rootdelete.sh remote sharedvar`. If the `ocr.loc` file is not on a shared file system, then execute the command `<CRS home>/install/rootdelete.sh remote nosharedvar`.

If you are deleting more than one node from your cluster, then repeat this step on each node that you are deleting.

6. Run `<CRS Home>/install/rootdeletenode.sh` on any remaining node in the cluster to delete the nodes from the Oracle cluster and to update the Oracle Cluster Registry (OCR). If you are deleting multiple nodes, then execute the command `<CRS Home>/install/rootdeletenode.sh node1, <node1-number>, node2, <node2-number>, ... <nodeN>, <node N-number>` where `node1` through `<nodeN>` is a list of the nodes that you want to delete, and `<node1-number>` through `<nodeN-number>` represents the node number. To determine the node number of any node, execute `<CRS Home>/bin/olsnodes -n`.
7. On the same node, execute the command `runInstaller -updateNodeList ORACLE_HOME=<Home location> CLUSTER_NODES=node1,node2,... <nodeN>` where `node1` through `<nodeN>` is a comma-separated list of nodes that are remaining in the cluster.
8. If you are not using a cluster file system, then on the node that you are deleting, remove the Oracle CRS software by executing the `rm` command. Make sure that you execute the `rm` command from the correct Oracle CRS home. Execute the `rm` command on every node that you are deleting.

Deleting Nodes from Oracle Clusters on Windows-Based Platforms

Execute the following procedures to delete nodes from Oracle clusters on Windows-based platforms:

1. If there are instances on the node that you want to delete, then execute the procedures in the section titled "[Deleting Instances from Real Application Clusters Databases](#)" on page 5-12 before continuing with these procedures. If you are deleting more than one node, then delete the instances from all the nodes that you are going to delete.
2. After you have removed all the instances from the node that you want to delete, run the command `%ORACLE_HOME%\bin\srvctl` on the node to delete the Oracle Cluster Ready Services (CRS) node applications. If you are deleting more than one node from your cluster, then execute the command `%ORACLE_HOME%\bin\srvctl remove nodeapps -n node1,node2,..., <nodeN>` where `node1` through `<nodeN>` is a comma-separated list of nodes that you want to delete.
3. If the node that you are deleting has an ASM instance, then delete the ASM instance using the `srvctl stop asm` and `srvctl remove asm` commands.
4. On the same node, execute the command `setup.exe -updateNodeList ORACLE_HOME=<Home location> CLUSTER_NODES=node1,node2,... <nodeN>` where `node1` through `<nodeN>` is a comma-separated list of the nodes that are remaining in the cluster.
5. On the same node, delete the Windows Registry entries and ASM services using Oradim.

6. On the nodes that you want to delete, if you are not using Oracle Cluster File System for the Oracle home, then remove the Oracle database software using the Windows administrative tools. Make sure that you select the correct Oracle home to delete.
7. From any remaining node in the cluster, disable the CRS applications by running the command `<CRS home>\bin\crssetup del -nn node1, <node1-number>, node2, <node2-number>, ... <nodeN>, <nodeN-number>` where `node1` through `<nodeN>` is a list of the nodes that you want delete, and `<node1-number>` through `<nodeN-number>` represents the node number. To determine the node number of any node, execute the command `<CRS Home>\bin\olsnodes -n`.
8. On the same node, execute the command `setup.exe -updateNodeList ORACLE_HOME=<Home location>CLUSTER_NODES=node1,node2,... <nodeN>` where `node1` through `<nodeN>` is a comma-separated list of the nodes that are remaining in the cluster.
9. On the nodes that you are deleting, remove the Oracle CRS software using Windows administrative tools. Make sure that you select the correct CRS home to delete. Also delete the CRS, CSS, and EVM services.

Configuring Recovery Manager and Archiving

This chapter explains how to configure Recovery Manager (RMAN) for use in Oracle Real Application Clusters (RAC) environments. This chapter also provides procedures for using RMAN for archiving in RAC environments and discusses redo logs and archived redo logs considerations. The topics in this chapter include:

- Overview of Configuring RMAN for Real Application Clusters
- Configuring the RMAN Snapshot Control File Location
- Configuring the RMAN Control File Autobackup Feature
- Managing Archived Redo Logs Using RMAN in Real Application Clusters
- Archived Redo Log File and Destination Conventions in RAC
- RMAN Archiving Configuration Scenarios
- Changing the Archiving Mode in Real Application Clusters

Overview of Configuring RMAN for Real Application Clusters

Recovery Manager (RMAN) enables you to back up, copy, restore, and recover datafiles, control files, server parameter files (SPFILES) and archived redo logs. It is included with the Oracle server and does not require separate installation. You can run RMAN from the command line or use RMAN in the Backup Manager in Oracle Enterprise Manager.

Configuring the RMAN Snapshot Control File Location

The snapshot control file is a temporary snapshot control file that RMAN creates to resynchronize from a read-consistent version of the control file. RMAN only needs a snapshot control file when resynchronizing with the recovery catalog or when making a backup of the current control file. In RAC, the snapshot control file is only needed on nodes where RMAN performs backups.

You can specify a cluster file system file or a raw device destination for the location of your snapshot control file so that this file is shared across all nodes in the cluster. You must ensure that this file physically or logically exists on all nodes that are used for backup and restore activities. Run the following RMAN command to determine the configured location of the snapshot control file:

```
SHOW SNAPSHOT CONTROLFILE NAME;
```

If needed, you can change the configured location of the snapshot control file. For example, on UNIX-based systems you can specify the snapshot control file location as `$ORACLE_HOME/dbs/scf/snap_prod.cf` by entering the following at the RMAN prompt:

```
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '$ORACLE_HOME/dbs/scf/snap_prod.cf';
```

This command globally sets the configuration for the location of the snapshot control file throughout your cluster database. Therefore, ensure that the directory `$ORACLE_HOME/dbs/scf` exists on all nodes that perform backups.

The `CONFIGURE` command creates persistent settings across RMAN sessions. Therefore, you do not need to run this command again unless you want to change the location of the snapshot control file. You can also specify a cluster file system file or a raw device destination for the location of your snapshot control file. This file is shared across all nodes in the cluster just like other datafiles in RAC. Refer to *Oracle Database Recovery Manager Reference* for more information about configuring the snapshot control file.

Configuring the RMAN Control File Autobackup Feature

If you set `CONFIGURE CONTROLFILE AUTOBACKUP` to `ON`, then RMAN automatically creates a control file and a SPFILE backup after you run the `BACKUP` or `COPY` commands. RMAN can also automatically restore an SPFILE if this is required to start an instance to perform recovery. This means that the default location for the SPFILE must be available to all nodes in your RAC database.

These features are important in disaster recovery because RMAN can restore the control file even without a recovery catalog. RMAN can restore an autobackup of the control file even after the loss of both the recovery catalog and the current control file. You can change the default name that RMAN gives to this file with the `CONFIGURE CONTROLFILE AUTOBACKUP FORMAT` command. Note that if you specify an absolute path name in this command, then this path must exist identically on all nodes that participate in backups.

RMAN performs the control file autobackup on the first allocated channel. Therefore, when you allocate multiple channels with different parameters, especially when you allocate a channel with the `CONNECT` command, determine which channel will perform the control file autobackup. Always allocate the channel for this node first. Refer to the *Oracle Database Backup and Recovery Advanced User's Guide* for more information about using the control file autobackup feature.

Managing Archived Redo Logs Using RMAN in Real Application Clusters

After configuring the RMAN snapshot control file location and enabling the RMAN control file autobackup feature, you can decide how to configure your environment to manage the archived redo logs that each node generates. When a node generates an archived redo log, Oracle always records the filename of the log in the control file of the target database. If you are using a recovery catalog, then RMAN also records the archived redo log filenames in the recovery catalog when a resynchronization occurs.

The archived redo log naming scheme that you use is important because when a node writes to a log with a specific filename on its file system, the file must be readable by any node that needs to access this archived redo log. For example, if node 1 archives a log to `/oracle/arc_dest/log_1_100.arc`, then node 2 can only back up this archived redo log if it can read `/oracle/arc_dest/log_1_100.arc` on its own file system.

The backup and recovery strategy that you choose depends on how you configure the archiving destinations for each node. Whether only one node performs archived redo log backups or all nodes perform archived redo log backups, you need to ensure that all archived redo logs are backed up. Because only one instance can perform recovery, the node of the instance performing recovery must have read access to all archived redo logs in your cluster.

Guidelines and Considerations for Archived Redo Logs

The primary consideration is to ensure that all archived redo logs can be read from every node during recovery, and if possible during backups. This section illustrates the archived redo log naming issues for configuring archiving in your cluster database. The scenario described here is for a non-cluster file system archiving scheme. Assume that the following conditions are met:

- Configure each node to write to a local archiving directory that is named the same on each node.
- Do *not* set up a cluster file system (in other words, each node can only read from and write to its own local file system). Refer to information about cluster file systems later in this chapter.
- Do not use NFS or mapped drives to enable the nodes in the cluster to gain read/write access to one another.

Example 6–1 Example Configuration for the initialization parameters file

```
sid1.log_archive_dest_1 = (location=/arc_dest)
sid2.log_archive_dest_1 = (location=/arc_dest)
sid3.log_archive_dest_1 = (location=/arc_dest)
```

Assume that the filenames of the archived redo logs are recorded in the control file as follows:

```
/arc_dest/log_1_62.arc
/arc_dest/log_2_100.arc
```

```

/arc_dest/log_2_101.arc
/arc_dest/log_3_70.arc
/arc_dest/log_1_63.arc

```

Given this scenario, assume that your RAC database performs recovery. If node 1 tries to read the logs for recovery, then it searches its local `/arc_dest` directory for the filenames as they are recorded in the control file. Using this data, node 1 only finds the logs that it archived locally, for example `/arc_dest/log_1_62.arc` and `/arc_dest/log_1_63.arc`. However, node 1 cannot apply the other logs because the filenames for these logs are not readable on its local file system. Thus, the recovery stalls. Avoid this by implementing the naming conventions described in the next section and then configure your cluster according to the scenarios described in ["RMAN Archiving Configuration Scenarios"](#) on page 6-4.

Archived Redo Log File and Destination Conventions in RAC

For any archived redo log configuration, uniquely identify the archived redo logs with the `LOG_ARCHIVE_FORMAT` parameter. The format of this parameter is operating system-specific and it can include text strings, one or more variables, and a filename extension.

Table 6–1 Archived Redo Log Filename Format Parameters

Parameter	Description	Example
<code>%T</code>	Thread number, left-zero-padded	<code>log_0001</code>
<code>%t</code>	Thread number, not padded	<code>log_1</code>
<code>%S</code>	Log sequence number, left-zero-padded	<code>log_0000000251</code>
<code>%s</code>	Log sequence number, not padded	<code>log_251</code>

The thread parameters `%t` or `%T` are mandatory for RAC. For example, if the instance associated with redo thread number 1 sets `LOG_ARCHIVE_FORMAT` to `log_%t_%s.arc`, then its archived redo log files are named:

```

log_1_1000.arc
log_1_1001.arc
log_1_1002.arc
.
.
.

```

See Also: *Oracle Database Administrator's Guide* about specifying the archived redo log filename format and destination, and Oracle platform-specific documentation about the default log archiving format and destination

RMAN Archiving Configuration Scenarios

This section describes the archiving scenarios for a RAC database. The two configuration scenarios in this chapter describe a three-node UNIX cluster for a RAC database. For both scenarios, the `LOG_ARCHIVE_FORMAT` that you specify for the instance performing recovery must be the same as the format that you specified for the instances that archived the files.

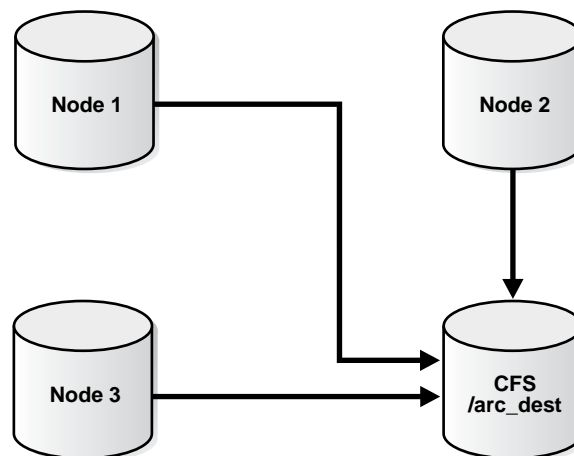
Cluster File System Archiving Scheme

The preferred configuration for RAC is to use 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.

In this case., each node writes to a single cluster file system archived redo log destination and can read the archived redo log files of the other nodes. Read access is achieved for all nodes with a cluster file system. For example, if node 1 archives a log to `/arc_dest/log_1_100.arc` on the cluster file system, then any other node in the cluster can also read this file.

If you do not use a cluster file system, then the archived redo log files cannot be on raw devices. This is because raw devices do not allow sequential writing of consecutive archive log files.

Figure 6–1 Cluster File System Archiving Scheme



Advantages and Disadvantages of the Cluster File System Archiving Scheme

The advantage of this scheme is that none of the nodes uses the network to archive logs. If each node has a local tape drive, then you can distribute an archived redo log backup so that each node backs up local logs without accessing the network. Because the filename written by a node can be read by any node in the cluster, RMAN can back up all logs from any node in the cluster. Backup and restore scripts are simplified because each node has access to all archived redo logs.

Initialization Parameter Settings for the Cluster File System Archiving Scheme

In the cluster file system scheme, each node archives to a directory that is identified with the same name on all instances within the cluster database. To configure this, set values for the `LOG_ARCH_DEST_n` parameter for each instance using the `sid` designator as in the following example:

```

sid1.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
sid2.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
sid3.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest"
  
```

The following list shows archived redo log entry examples that would appear in the RMAN catalog or the in the control file based on the previous example. Note that any node can archive logs using any of the threads:

```
/arc_dest/log_1_999.arc  
/arc_dest/log_1_1000.arc  
/arc_dest/log_1_1001.arc <- thread 1 archived in node 3  
/arc_dest/log_3_1563.arc <- thread 3 archived in node 2  
/arc_dest/log_2_753.arc <- thread 2 archived in node 1  
/arc_dest/log_2_754.arc  
/arc_dest/log_3_1564.arc
```

Location of Archived Logs for the Cluster File System Archiving Scheme

Because the file system is shared and because each node is writing its archived redo logs to directory `/arc_dest` in the cluster file system, each node can read the logs written by itself as well as any other node.

Non-Cluster File System Local Archiving Scheme

In the non-cluster file system local archiving scheme, each node archives to a uniquely named local directory. If recovery is required, then you can configure the recovery node so that it can access directories on the other nodes remotely. For example, use NFS on UNIX-based systems or shared drives on Windows-based systems. Therefore, each node writes only to a local destination, but each node can also read archived redo log files in remote directories on the other nodes.

To enable RMAN to back up and recover a RAC database in one step, all archived redo logs must have uniquely identifiable names throughout the cluster. To do this, however, you cannot use the technique described in "[Cluster File System Archiving Scheme](#)" on page 6-5 to have more than one node archive to a directory such as `/arc_dest`. In UNIX environments only, the archived redo log files cannot be on the shared disk because UNIX shared disks are raw devices that you cannot easily partition for use with archived redo logs.

Advantages and Disadvantages for Non-Cluster File System Local Archiving

The advantage of this scheme is that if each node has a local tape drive, then you can distribute an archived redo log backup so that each node backs up local logs without accessing the network. The disadvantage of this scheme is that for media recovery, you must configure the node performing recovery for remote access so that it can read the archived redo log files in the archiving directories on the other nodes.

If only one node has a local tape drive, then you cannot back up all logs from a single node without configuring NFS or manually transferring the logs. This scheme has a single point of failure. If one node fails after the most recent backup, then the archived redo logs on this node that were generated *after* the backup are lost.

If you are in a recovery situation and if you do not have all the available archive logs, then you must perform an incomplete recovery up to the first missing archived redo log sequence number. You do not have to use a specific configuration for this scheme. However, if you want to distribute the backup processing onto multiple nodes, the easiest method is to configure channels as described in the backup scenarios in [Chapter 7, "Managing Backup and Recovery"](#).

Initialization Parameter Settings for Non-Cluster File System Local Archiving

You can set the archiving destination values as follows in the initialization parameter file:

```
sid1.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_1"  
sid2.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_2"  
sid3.LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest_3"
```


The following list shows the possible archived redo log entries in the database control file. Note that any node is able to archive logs from any of the threads:

```

/arc_dest_1/log_1_1000.arc
/arc_dest_2/log_1_1001.arc <- thread 1 archived in node 2
/arc_dest_2/log_3_1563.arc <- thread 3 archived in node 2
/arc_dest_1/log_2_753.arc <- thread 2 archived in node 1
/arc_dest_2/log_2_754.arc
/arc_dest_3/log_3_1564.arc
    
```

Location of Archived Logs for Non-Cluster File System Local Archiving

As illustrated in [Table 6–2](#), each node has a directory containing the locally archived redo logs. Additionally, if you mount directories on the other nodes remotely through NFS or shared drives, then each node has two remote directories through which RMAN can read the archived redo log files that are archived by the remaining nodes.

Table 6–2 Location of Logs for Non-Cluster File System Local Archiving

Node ...	Can read the archived redo log files in the directory ...	For logs archived by node ...
1	/arc_dest_1	1
1	/arc_dest_2	2 (through NFS)
1	/arc_dest_3	3 (through NFS)
2	/arc_dest_1	1 (through NFS)
2	/arc_dest_2	2
2	/arc_dest_3	3 (through NFS)
3	/arc_dest_1	1 (through NFS)
3	/arc_dest_2	2 (through NFS)
3	/arc_dest_3	3

File System Configuration for Non-Cluster File System Local Archiving

Because NFS is not required to perform backups, node 1 can back up its local logs to its tape drive, node 2 can back up its local logs to its tape drive, and so on. However, if you are performing recovery and a surviving instance must read all the logs that are on disk but not yet backed up, then you should configure NFS as shown in [Table 6–3](#).

Table 6–3 NFS Configuration for Shared Read Local Archiving

Node	Directory ...	Is configured ...	And mounted on ...	On node ...
1	/arc_dest_1	Local read/write	n/a	n/a
1	/arc_dest_2	NFS read	/arc_dest_2	2
1	/arc_dest_3	NFS read	/arc_dest_3	3
2	/arc_dest_1	NFS read	/arc_dest_1	1
2	/arc_dest_2	Local read/write	n/a	n/a
2	/arc_dest_3	NFS read	/arc_dest_3	3
3	/arc_dest_1	NFS read	/arc_dest_1	1
3	/arc_dest_2	NFS read	/arc_dest_2	2
3	/arc_dest_3	Local read/write	n/a	n/a

Changing the Archiving Mode in Real Application Clusters

To change the archiving mode in a RAC environment, the database must be mounted (not open) by an exclusive instance. In other words, set the `CLUSTER_DATABASE` parameter to `FALSE`. After executing the `ALTER DATABASE SQL` statement to change the archive log mode, shutdown the instance and restart it with the `CLUSTER_DATABASE` parameter reset to `TRUE` before you restart the other instances. When the database goes into archive log mode, the ARCH processes automatically start.

Monitoring the Archiver Processes

After your RMAN configuration is operative in your RAC environment, use the `GV$ARCHIVE_PROCESSES` and `V$ARCHIVE_PROCESSES` views to determine the status of the archiver processes. Depending on whether you query the global or local views, these views display information for all database instances, or for only the instance to which you are connected. Refer to *Oracle Database Reference* for more information about the database views.

Managing Backup and Recovery

This chapter explains instance recovery and how to use Recovery Manager (RMAN) to back up and restore Oracle Real Application Cluster (RAC) databases. This chapter also describes RAC instance recovery as well as parallel backup and recovery with SQL*Plus. The topics in this chapter include:

- [Instance Recovery in Real Application Clusters](#)
- [RMAN Backup Scenarios for Real Application Clusters](#)
- [Media Recovery in Real Application Clusters](#)
- [RMAN Restore Scenarios for Real Application Clusters](#)
- [RMAN Recovery Through Resetlogs in Real Application Clusters](#)
- [Parallel Recovery in Real Application Clusters](#)
- [Using a Flash Recovery Area in Real Application Clusters](#)

Instance Recovery in Real Application Clusters

Instance failure occurs when software or hardware problems disable an instance. After instance failure, Oracle automatically uses the online redo logs to perform recovery as described in this section.

Single Node Failure in Real Application Clusters

Instance recovery in RAC does not include the recovery of applications that were running on the failed instance. Oracle clusterware restarts the instance automatically. You can also use callout programs as described in the example in the *Oracle Real Application Clusters Deployment and Performance Guide* to trigger application recovery.

Applications that were running continue by using failure recognition and recovery. This provides consistent and uninterrupted service in the event of hardware or software failures. When one instance performs recovery for another instance, the surviving instance reads online redo logs generated by the failed instance and uses that information to ensure that committed transactions are recorded in the database. Thus, data from committed transactions is not lost. The instance performing recovery rolls back transactions that were active at the time of the failure and releases resources used by those transactions.

Note: All online redo logs must be accessible for instance recovery. Therefore, Oracle recommends that you mirror your online redo logs.

Multiple-Node Failures in Real Application Clusters

When multiple node failures occur, as long as one instance survives, RAC performs instance recovery for any other instances that fail. If all instances of a RAC database fail, then Oracle automatically recovers the instances the next time one instance opens the database. The instance performing recovery can mount the database in either shared or exclusive mode from any node of a RAC database. This recovery procedure is the same for Oracle running in shared mode as it is for Oracle running in exclusive mode, except that one instance performs instance recovery for all the failed instances.

Using RMAN to Create Backups in Real Application Clusters

Oracle provides RMAN for backing up and restoring the database. RMAN enables you to back up, copy, restore, and recover datafiles, control files, SPFILEs, and archived redo logs. RMAN is included with the Oracle server and it is installed by default. You can run RMAN from the command line or you can use it from the Backup Manager in Oracle Enterprise Manager. The procedures for using RMAN in RAC environments do not differ substantially from those for Oracle single-instance environments. Refer to the Oracle Backup and Recovery documentation set for more information about single-instance RMAN backup procedures.

Using RMAN to Restore the Server Parameter File (SPFILE)

RMAN can restore the server parameter file either to the default location or to a location that you specify. This procedure is described in *Oracle Database Backup and Recovery Basics*.

RMAN and Oracle Net in Real Application Clusters

You cannot specify a net service name that uses Oracle Net to distribute RMAN connections to more than one instance. In any RMAN connection made through a net service name, each net service name must specify only one instance. This applies to all RMAN connections, whether from the command line or through the `CONNECT` clause in `ALLOCATE CHANNEL` or `CONFIGURE CHANNEL RMAN` commands. Additionally, when you use RMAN from the command line, you can only connect to one instance in a RAC database at a time. Assume that `node1`, `node2`, and `node3` are net service names for three instances in a RAC configuration. In this case, connect to the target database with only one of these net service names, for example:

```
% rman TARGET SYS/oracle@node2 CATALOG rman/cat@catdb
```

Channel Connections to Cluster Instances

When making backups using channels connected to different instances, each allocated channel can connect to a different instance in the cluster, and each channel connection must resolve to only one instance. For example, configure automatic channels as follows:

```
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT = 'SYS/oracle@node1';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT = 'SYS/oracle@node2';
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT = 'SYS/oracle@node3';
```

During a backup, the instances to which the channels connect must be either all mounted or all open. For example, if the `node1` instance has the database mounted while the `node2` and `node3` instances have the database open, then the backup fails.

Node Affinity Awareness of Fast Connections

In some cluster database configurations, some nodes of the cluster have faster access to certain datafiles than to other datafiles. RMAN automatically detects this, which is known as node affinity awareness. When deciding which channel to use to back up a particular datafile, RMAN gives preference to the nodes with faster access to the datafiles that you want to back up. For example, if you have a three-node cluster, and if node 1 has faster read/write access to datafiles 7, 8, and 9 than the other nodes, then node 1 has greater node affinity to those files than nodes 2 and 3.

To use node affinity, configure RMAN channels on the nodes of the cluster that have affinity to the datafiles you want to back up. For example, use the syntax:

```
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'user1/password1@node1';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'user2/password2@node2';
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT 'user3/password3@node3';
```

Refer to *Oracle Database Recovery Manager Reference* for more information about the `CONNECT` clause of the `CONFIGURE CHANNEL` statement.

Readability of Files To Be Backed Up

The node performing the backup must be able to read all of the files specified in the `BACKUP` command. For example, assume that you run the following command on node 1 of a three-node cluster:

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

In this case, RMAN attempts to back up all datafiles, archived redo logs, and SPFILEs. Because the datafiles are either cluster file system files or files on a shared disk, RMAN can read them. However, RMAN cannot back up any of the archived redo logs that the local node cannot read. The archiving scenarios in [Chapter 6, "Configuring Recovery Manager and Archiving"](#) explain how to configure the environment so that all archived redo logs are accessible by the node performing the backup.

Deleting Archived Redo Logs After a Successful Backup

The BACKUP command must be able to delete the archived redo logs from disk after backing them up. The following script is an example of one method for deleting the archived redo logs from each node after backing them up:

```
ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT 'SYS/oracle@node1';
DELETE ARCHIVELOG LIKE '%arc_dest_1%'
  BACKED UP 1 TIMES TO DEVICE TYPE sbt;
RELEASE CHANNEL;

ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT 'SYS/oracle@node2';
DELETE ARCHIVELOG LIKE '%arc_dest_2%'
  BACKED UP 1 TIMES TO DEVICE TYPE sbt;
RELEASE CHANNEL;

ALLOCATE CHANNEL FOR MAINTENANCE DEVICE TYPE DISK CONNECT 'SYS/oracle@node3';
DELETE ARCHIVELOG LIKE '%arc_dest_3%'
  BACKED UP 1 TIMES TO DEVICE TYPE sbt;
RELEASE CHANNEL;
```

Distribution of Backups

When configuring the backup options for RAC, you have three possible configurations:

- **Network Backup Server.** A dedicated backup server performs and manages backups for the cluster and the cluster database. None of the nodes have local backup appliances.
- **One Local Drive.** One node has access to a local backup appliance and performs and manages backups for the cluster database. All nodes of the cluster should be on a cluster file system to be able to read all datafiles, archived redo logs, and SPFILEs. Oracle recommends that you do not use the non-cluster file system archiving scheme if you have backup media on only one local drive.
- **Multiple Drives.** Each node has access to a local backup appliance and can write to its own local backup media.

In the cluster file system scheme, any node can access all the datafiles, archived redo logs, and SPFILEs. In the non-cluster file system scheme, you must write the backup script so that the backup is distributed to the correct drive and path for each node. For example, node 1 can back up the archived redo logs whose path names begin with `/arc_dest_1`, node 2 can back up the archived redo logs whose path names begin with `/arc_dest_2`, and node 3 can back up the archived redo logs whose path names begin with `/arc_dest_3`.

Autolocation for Backup and Restore Commands

RMAN automatically performs autolocation of all files that it needs to back up or restore. This feature is automatically enabled whenever the allocated channels use different CONNECT or PARMS settings.

If you use the non-cluster file system local archiving scheme, then a node can only read the archived redo logs that were generated by an instance on that node. RMAN never attempts to back up archived redo logs on a channel it cannot read.

During a restore operation, RMAN automatically performs the autolocation of backups. A channel connected to a specific node only attempts to restore files that were backed up to the node. For example, assume that log sequence 1001 is backed up to the drive attached to node 1, while log 1002 is backed up to the drive attached to node 2. If you then allocate channels that connect to each node, then the channel connected to node 1 can restore log 1001 (but not 1002), and the channel connected to node 2 can restore log 1002 (but not 1001).

RMAN Backup Scenarios for Real Application Clusters

This section describes the options for backup scenarios.

Cluster File System Backup Scheme

In a cluster file system backup scheme, each node in the cluster has read access to all the datafiles, archived redo logs, and SPFILES. This includes Automated Storage Management (ASM), cluster file systems, and Network Attached Storage (NAS).

Backing Up to One Local Drive in the Cluster File System Archiving Scheme

This scheme assumes that only one node in the cluster has a local backup appliance such as a tape drive. In this case, run the following one-time configuration commands:

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 1;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
```

Because any node performing the backup has read/write access to the archived redo logs written by the other nodes, the backup script for any node is simple:

```
BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

In this case, the tape drive receives all datafiles, archived redo logs, and SPFILES.

Backing Up to Multiple Drives in the Cluster File System Archiving Scheme

This scheme assumes that each node in the cluster has its own local tape drive. Perform the following one-time configuration so that one channel is configured for each node in the cluster. This is a one-time configuration step. For example, enter the following at the RMAN prompt:

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE DEFAULT DEVICE TYPE TO sbt;
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'user1/password1@node1';
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'user2/password2@node2';
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT 'user3/password3@node3';
```

Similarly, you can perform this configuration for a device type of DISK. The following backup script, which you can run from any node in the cluster, distributes the datafiles, archived redo logs, and SPFILE backups among the backup drives:

```
BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

For example, if the database contains 10 datafiles and 100 archived redo logs are on disk, then the node 1 backup drive can back up datafiles 1, 3, and 7 and logs 1-33.

Node 2 can back up datafiles 2, 5, and 10 and logs 34-66. The node 3 backup drive can back up datafiles 4, 6, 8 and 9 as well as archived redo logs 67-100.

Non-Cluster File System Backup Scheme

In a non-cluster file system environment, each node can back up only its own local archived redo logs. For example, node 1 cannot access the archived redo logs on node 2 or node 3 unless you configure the network file system for remote access. To configure NFS, distribute the backup to multiple drives. However, if you configure NFS for backups, then you can only back up to one drive.

Backing Up to Multiple Drives in a Non-Cluster File System Backup Scheme

This scheme assumes that each node in the cluster has its own local tape drive. Perform the following one-time configuration to configure one channel for each node in the cluster. For example, enter the following at the RMAN prompt:

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;  
CONFIGURE DEFAULT DEVICE TYPE TO sbt;  
CONFIGURE CHANNEL 1 DEVICE TYPE sbt CONNECT 'user1/password1@node1';  
CONFIGURE CHANNEL 2 DEVICE TYPE sbt CONNECT 'user2/password2@node2';  
CONFIGURE CHANNEL 3 DEVICE TYPE sbt CONNECT 'user3/password3@node3';
```

Similarly, you can perform this configuration for a device type of DISK.

Develop a production backup script for whole database backups that you can run from any node. The RMAN autolocation feature ensures that the channel allocated on each node only backs up the archived redo logs that are located on that node. The following example uses automatic channels to make a database and archived redo log backup:

```
BACKUP DATABASE PLUS ARCHIVELOG DELETE INPUT;
```

In this example, the datafile backups, archived redo logs, and SPFILE backups are distributed among the different tape drives. However, channel 1 can only read the logs archived locally on `/arc_dest_1`. This is because the autolocation feature restricts channel 1 to only back up the archived redo logs in the `/arc_dest_1` directory and because node 2 can only read files in the `/arc_dest_2` directory, channel 2 can only back up the archived redo logs in the `/arc_dest_2` directory, and so on. The important point is that *all* logs are backed up, but they are distributed among the different drives.

Media Recovery in Real Application Clusters

Media recovery must be user-initiated through a client application, whereas instance recovery is automatically performed by the database. In these situations, use RMAN to restore backups of the datafiles and then recover the database. The procedures for RMAN media recovery in RAC environments do not differ substantially from the media recovery procedures for single-instance environments. The node that performs the recovery must be able to restore all the required datafiles. That node must also be able to either read all the required archived redo logs on disk or be able to restore them from backups.

RMAN Restore Scenarios for Real Application Clusters

This section describes the RMAN restore scenarios.

Cluster File System Restore Scheme

The restore and recovery procedures in a cluster file system scheme do not differ substantially from Oracle single-instance scenarios.

Restore and Recovery of Nondistributed Backups

First, refer to ["Backing Up to One Local Drive in the Cluster File System Archiving Scheme"](#) on page 7-5 to perform the one-time configuration.

In this example, assume that node 3 performs the backups. If node 3 is available for the restore and recovery processing, and if all the existing archived redo logs have been backed up or are on disk, then run the following commands to perform complete recovery:

```
RESTORE DATABASE;  
RECOVER DATABASE;
```

If node 3 performed the backups but is unavailable, then configure a media management device for one of the remaining nodes and make the backup media from node 3 available to this device.

Restore and Recovery of Distributed Backups in a Cluster File System Scheme

First, refer to ["Backing Up to Multiple Drives in the Cluster File System Archiving Scheme"](#) on page 7-5 to perform the one-time configuration so that one channel is configured for each node in the cluster. If all existing archived redo logs have been backed up or are on disk, then run the following commands for complete recovery from any node in the cluster:

```
RESTORE DATABASE;  
RECOVER DATABASE;
```

Because RMAN autolocates the backups before restoring them, the channel connected to each node only restores the files that were backed up to the tape drive attached to the node.

Non-Cluster File System Restore Scheme

In this scheme, each node archives locally to a different directory. For example, node 1 archives to `/arc_dest_1`, node 2 archives to `/arc_dest_2`, and node 3 archives to `/arc_dest_3`. You must configure NFS so that the recovery node can read the archiving directories on the remaining nodes. The restore and recovery procedure depends on whether the backups are distributed or nondistributed.

Restore and Recovery of Distributed Backups in a Non-Cluster File System Archiving Scheme

First, refer to ["Backing Up to Multiple Drives in a Non-Cluster File System Backup Scheme"](#) on page 7-6. If all nodes are available and if all archived redo logs have been backed up, then you can perform a complete restore and recovery by mounting the database and running the following commands from any node:

```
RESTORE DATABASE;  
RECOVER DATABASE;
```

The recovery node begins a server session on each node in the cluster. Because this example assumes that database backups are distributed, the server sessions restore the backup datafiles from the tape drives attached to each node. Because the NFS configuration enables each node read access to the other nodes, the recovery node can read and apply the archived redo logs located on the local and remote disks. No manual transfer of archived redo logs is required.

RMAN Recovery Through Resetlogs in Real Application Clusters

The *resetlogs* operation automatically archives online logs. This ensures that your database has the necessary archived redo logs if recovery was done to a point in time in online or standby logs. You do not need to perform a full backup after a resetlogs operation.

The default archive log format includes a resetlogs id. There is no need to change backup scripts after using resetlogs. RMAN backs up earlier incarnations of logs on running `BACKUP ARCHIVELOG ALL` or `BACKUP ARCHIVELOG FROM TIME` or `BACKUP ARCHIVELOG FROM SCN` command.

Parallel Recovery in Real Application Clusters

You can use parallel instance recovery, parallel failure recovery, and parallel media recovery in RAC databases. Refer to *Oracle Database Backup and Recovery Advanced User's Guide* for more information on these topics.

Parallel Recovery with RMAN

With RMAN's `RESTORE` and `RECOVER` commands, Oracle automatically makes parallel the following three stages of recovery:

Restoring Datafiles When restoring datafiles, the number of channels you allocate in the RMAN recover 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 RMAN applies archived redo logs using a specific number of parallel processes as determined by the setting for the `RECOVERY_PARALLELISM` initialization parameter. This is described under the topic "[Setting the RECOVERY_PARALLELISM Parameter](#)" on page 7-8.

Media Recovery Parallelism Media recovery parallelism is controlled by the `PARALLEL` clause of the `ALTER DATABASE RECOVER` statement.

Parallel Recovery with SQL*Plus

If you have user-managed methods to back up and recover your database, then you can parallelize instance and media recovery using either of the procedures described in this section.

Setting the RECOVERY_PARALLELISM Parameter The `RECOVERY_PARALLELISM` initialization parameter specifies the number of processes that participate in instance and crash recovery.

One process reads the archived redo log files sequentially and dispatches redo information to several recovery processes. The recovery processes then apply the changes from the archived redo log files to the datafiles. A value of 0 or 1 indicates that Oracle performs recovery serially by one process. The value of this parameter cannot exceed the value of the `PARALLEL_MAX_SERVERS` parameter.

On multiple-CPU systems, the default for instance, crash, and media recovery is to operate in parallel mode. You can, however, enforce the use of serial recovery by using either the `RECOVERY_PARALLELISM` parameter or the `NOPARALLEL` clause of the `ALTER DATABASE STATEMENT`.

Specifying RECOVER Statement Options When you use the `RECOVER` statement to parallelize instance and media recovery, the allocation of recovery processes to instances is operating system-specific. The `DEGREE` keyword of the `PARALLEL` clause can either signify the number of processes on each instance of a RAC database or the number of processes to distribute across all instances.

Using a Flash Recovery Area in Real Application Clusters

To use a flash recovery area in RAC, you must place it on an ASM disk, a Cluster File System, or on a shared directory that is configured through NFS for each RAC instance. In other words, the flash recovery area must be shared among all the instances of a RAC database. In addition, set the parameter `DB_RECOVERY_FILE_DEST` to the same value on all instances.

Administrative Options

This chapter describes administrative tasks or options within Oracle tools that are specific to Oracle Real Application Clusters (RAC) and not discussed elsewhere in this book. In some cases, you have a choice of tools to perform a task while other tasks must be executed through a specific tool, such as Enterprise Manager or SRVCTL. The topics in this chapter are:

- [Optional Enterprise Manager Tasks in Real Application Clusters](#)
- [Real Application Clusters Administration Procedures for Enterprise Manager](#)
- [Additional Information About SQL*Plus in Real Application Clusters](#)
- [Quiescing Real Application Clusters Databases](#)
- [Administering System and Network Interfaces with the OIFCFG \(Oracle Interface Configuration\) Tool](#)

See Also: *Oracle Enterprise Manager Concepts* and the Enterprise Manager online help for more information about Enterprise Manager

Optional Enterprise Manager Tasks in Real Application Clusters

Within Enterprise Manager, RAC-specific administrative tasks generally focus on two levels: tasks that affect an entire cluster database and tasks that affect specific instances. For example, you can use Enterprise Manager to administer storage, the schema, and security at the cluster database level. Or you can perform instance-specific commands such as setting parameters or creating resource plans.

Because there is one Enterprise Manager Agent on each node of a RAC database, you can use any URL for that database to administer it with Enterprise Manager. You can manage all of the following RAC components as targets in your Enterprise Manager framework:

- Host cluster—Accessible from the Cluster Database Home Page.
- Cluster database instances—Links to the instance pages appear on the Cluster Database Home Page.
- Hosts and listeners—Links to hosts and listeners appear on the Cluster Database Instance Pages.

See Also: *Oracle Enterprise Manager Concepts* for information about creating administrator accounts, using privileges, defining roles, and so on, and *Oracle Enterprise Manager Advanced Configuration* for information about configuring Enterprise Manager Grid Control for multiple RAC databases

Using Enterprise Manager to Discover Nodes and Instances

Enterprise Manager automatically detects changes in your configuration, for example, if you add nodes or instances. Enterprise Manager rediscovers pre-existing instances as well as newly added instances. To do this, use Enterprise Manager Grid Control as follows:

1. Log in to Enterprise Manager Database Control and click the Targets tab.
2. Click a database name to view all of the available targets. The column labeled Types indicates a RAC database with the entry "Cluster Database".
3. Add the database target by clicking Add on the right-hand side of the page.
4. Click the flashlight icon to display the available host names.
5. Select a host, click Continue, and Enterprise Manager performs discovery to locate and display the cluster database and its associated instances.

If you cannot discover your reconfigured cluster database and all of its instances using this procedure, then manually configure your targets for Enterprise Manager by adding the Oracle home, SID, instances, and so on.

See Also: *The Oracle Real Application Clusters Deployment and Performance Guide* for more information about using Enterprise Manager to monitor RAC databases

Enterprise Manager Pages for Real Application Clusters

This section describes the following Enterprise Manager pages for RAC:

- [Databases Summary Page](#)
- [Cluster Database Home Page](#)

- [Cluster Database Instances Pages](#)
- [The Databases Overview Page for Real Application Clusters](#)
- [The Cluster Home Page for Real Application Clusters](#)

Databases Summary Page

This is a top-level page that shows cluster and single-instance databases. If there are cluster databases in your environment, the Databases Summary page displays "Cluster Database" in the Type column. The page also indicates cluster database availability as well as the ratio of active instances to inactive instances. Click a cluster database link and Enterprise Manager displays the Cluster Database Home Page for that database, which is described under the following heading.

Cluster Database Home Page

From the Cluster Database Home Page you can manage the cluster nodes and hosts as well as cluster subcomponents such as instances and listeners. The Cluster Database Home Page is also a summary page for cluster database management that provides an overview of cluster database activity. Enterprise Manager uses a unique database name to identify the cluster database it represents. You can use the Administration tab on this page to perform many activities such as:

- Create undo tablespaces and redo threads and assign them to specific instances, SPFILE, create a backup
- Start, stop, and relocate database services at the cluster database level

You can use the Maintenance tab on this page to perform operations such as:

- Create backup and recovery scenarios
- Toggle the archive mode on and off
- Administer recovery settings
- Manage resource plans for the database and its instances

Cluster Database Instances Pages

Instances pages show instance-specific information similar to the information that you would see on a single-instance Oracle database.

The Databases Overview Page for Real Application Clusters

The Databases Overview Page links to Cluster Home Pages and to the node or instance Home Pages.

The Cluster Home Page for Real Application Clusters

The Cluster Home Page displays an overview of activities and detailed reports at both the cluster and instance levels. The Cluster Home Page has the following sections:

- General Section—Provides a cluster status overview.
- Configuration Section—Lists the hardware platform, operating system and version, and Oracle or vendor clusterware version.

- Cluster Databases Table—Displays the cluster databases associated with a cluster, their availability, and any cluster database alerts. You can access the individual Cluster Database Home Pages from the Cluster Databases Table.
- Alerts Table—Provides alert information such as severity rating.
- Hosts Table—Displays information about the hosts or nodes in the cluster.

Instance Pages for Real Application Clusters

The RAC-specific contents of Instance Pages are:

- Configuration—You can view instance states, view and edit initialization parameters at the instance level and at the cluster database level, and view resource plan performance statistics. You can also view and modify the undo tablespaces assigned to an instance and the undo tablespace retention period.
- Sessions—You can list the statuses of connected users, view the latest SQL for specific sessions, and terminate sessions.
- Locks—You can view details for currently held User type and System type locks.
- Resource Plans—You can define and modify the resource plans for the cluster database and also activate and deactivate resource plans for specific instances. You can also use the Resource Plan Schedule to schedule resource plan activation.

Real Application Clusters Administration Procedures for Enterprise Manager

The Cluster Database Home page shows all the instances in the RAC database and provides an aggregate collection of several RAC-specific statistics that are collected by the Automatic Workload Repository for server manageability.

You do not need to navigate to an instance-specific page to see these details. However, on the Cluster Database Home page, if an instance is down that should be operating, or if an instance has a high number of alerts, then you can drill down to the instance-specific page for each alert.

To perform specific administrative tasks as described in the remainder of this section, log in to the target RAC database, navigate to the Cluster Database Home page, and click the Administration tab.

Administering Enterprise Manager Jobs in Real Application Clusters

You can administer Enterprise Manager jobs at both the database and instance levels. For example, you can create a job at the cluster database level and the job will run on any active instance of the target RAC database. Or you can create a job at the instance level and the job will only run on the specific instance for which you created it. In the event of a failure, recurring jobs can run on a surviving instance.

Creating Enterprise Manager Jobs in Real Application Clusters

Because you can create jobs at the instance level, cluster level, or cluster database level, jobs can run on any available host in the cluster database. This applies to scheduled jobs as well. Enterprise Manager also displays job activity in several categories, namely, Active, History, and Library.

You use the Jobs tab to submit operating system scripts and SQL scripts and to examine scheduled jobs. For example, to create a backup job for a specific RAC database:

1. Click Targets and click the database for which you want to create the job.
2. Log in to the target database.
3. When Enterprise Manager displays the Database Home page, click Maintenance.
4. Complete the Enterprise Manage Job Wizard panels to create the job.

Administering Alerts in Real Application Clusters with Enterprise Manager

You can use Enterprise Manager to configure alerts for RAC environments. You can also configure specialized tests for RAC databases such as global cache converts, consistent read requests, and so on.

See Also: The *Oracle Real Application Clusters Deployment and Performance Guide* for more information about using Enterprise Manager to monitor RAC environments

Enterprise Manager distinguishes between database- and instance-level alerts in RAC environments. Alert thresholds for instance level alerts, such as archive log alerts, can be set at the instance target level. This enables you to receive alerts for the specific instance if performance exceeds your threshold. You can also configure alerts at the database level, such as setting alerts for tablespaces. This enables you to avoid receiving duplicate alerts at each instance.

See Also: The *Oracle Real Application Clusters Deployment and Performance Guide* for an example of configuring alerts in RAC and the *PL/SQL Packages and Types Reference* for information about using packages to configure thresholds

Performing Scheduled Maintenance Using Defined Blackouts in Enterprise Manager

You can define blackouts for all managed targets of a RAC database to prevent alerts from occurring while performing maintenance. You can define blackouts for an entire cluster database or for specific cluster database instances.

Additional Information About SQL*Plus in Real Application Clusters

The following sections describe the use of SQL*Plus in RAC environments:

- [How SQL*Plus Commands Affect Instances](#)
- [Verifying that Instances are Running](#)

How SQL*Plus Commands Affect Instances

Most SQL statements affect the current instance. You can use SQL*Plus to start and stop instances in the RAC database. You do not need to run SQL*Plus commands as root on UNIX-based systems or as Administrator on Windows-based systems. You need only the proper database account with the privileges that you normally use for a single-instance Oracle database. Some examples of how SQL*Plus commands affect instances are:

- The statement `ALTER SYSTEM SET CHECKPOINT LOCAL` only affects the instance to which you are currently connected, rather than the default instance or all instances.
- `ALTER SYSTEM CHECKPOINT LOCAL` affects the current instance.
- `ALTER SYSTEM CHECKPOINT` or `ALTER SYSTEM CHECKPOINT GLOBAL` affects *all* instances in the cluster database.
- `ALTER SYSTEM SWITCH LOGFILE` affects only the current instance.
 - To force a global log switch, use the `ALTER SYSTEM ARCHIVE LOG CURRENT` statement.
 - The `INSTANCE` option of `ALTER SYSTEM ARCHIVE LOG` enables you to archive each online redo log file for a specific instance.

[Table 8–1](#) describes how SQL*Plus commands affect instances.

Table 8–1 How SQL*Plus Commands Affect Instances

SQL*Plus Command	Associated Instance
ARCHIVE LOG	Always affects the current instance.
CONNECT	Affects the default instance if no instance is specified in the CONNECT command.
HOST	Affects the node running the SQL*Plus session, regardless of the location of the current and default instances.
RECOVER	Does not affect any particular instance, but rather the database.
SHOW INSTANCE	Displays information about the current instance, which can be different from the default local instance if you have redirected your commands to a remote instance.
SHOW PARAMETER and SHOW SGA	Displays parameter and SGA information from the current instance.
STARTUP and SHUTDOWN	Always affects the current instance. These are privileged SQL*Plus commands.

Verifying that Instances are Running

To verify that instances are running, on any node from a SQL*Plus prompt enter:

```
CONNECT SYS/password as SYSDBA
SELECT * FROM V$ACTIVE_INSTANCES;
```

Oracle returns output similar to the following:

```
INST_NUMBER INST_NAME
-----
1 db1-sun:db1
2 db2-sun:db2
3 db3-sun:db3
```

The output columns for this example are shown in [Table 8–2](#).

Table 8–2 Descriptions of V\$ACTIVE_INSTANCES Columns

Column	Description
INST_NUMBER	Identifies the instance number.
INST_NAME	Identifies the host name and instance name.

Quiescing Real Application Clusters Databases

The procedure for quiescing RAC databases is identical to quiescing a single-instance database. You use the `ALTER SYSTEM QUIESCE RESTRICTED` statement from one instance. You cannot open the database from any instance while the database is in the process of being quiesced. Once all non-DBA sessions become inactive, the `ALTER SYSTEM QUIESCE RESTRICTED` statement finishes, and the database is considered as in a quiesced state. In an Oracle RAC environment, this statement affects all instances, not just the one from which the statement is issued.

To successfully issue the `ALTER SYSTEM QUIESCE RESTRICTED` statement in a RAC environment, you must have the Database Resource Manager feature activated, and it must have been activated since instance startup for all instances in the cluster database. It is through the facilities of the Database Resource Manager that non-DBA sessions are prevented from becoming active. Also, while this statement is in effect, any attempt to change the current resource plan will be queued until after the system is unquiesced.

These conditions apply to RAC:

- If you issued the `ALTER SYSTEM QUIESCE RESTRICTED` statement but Oracle has not finished processing it, you cannot open the database.
- You cannot open the database if it is already in a quiesced state.
- The `ALTER SYSTEM QUIESCE RESTRICTED` and `ALTER SYSTEM UNQUIESCE` statements affect all instances in a RAC environment, not just the instance that issues the command.

Quiesced State and Cold Backups

You cannot use the quiesced state to take a cold backup. This is because Oracle background processes may still perform updates for Oracle internal purposes even while the database is in quiesced state. In addition, the file headers of online datafiles continue to look like they are being accessed. They do not look the same as if a clean shutdown were done. You can still take online backups while the database is in a quiesced state. Refer to the *Oracle Database Administrator's Guide* for details on the quiesce database feature and the *Oracle Database Reference* for more information about the `ALTER SYSTEM QUIESCE RESTRICTED` syntax.

Administering System and Network Interfaces with the OIFCFG (Oracle Interface Configuration) Tool

This section describes the following Oracle Interface Configuration (OIFCFG) topics:

- [Defining Network Interfaces with OIFCFG](#)
- [Syntax and Commands for the OIFCFG Command-Line Tool](#)

Use the OIFCFG command-line tool in single-instance Oracle databases and in RAC database environments to:

- Allocate and de-allocate network interfaces to components
- Direct components to use specific network interfaces
- Retrieve component configuration information

The Oracle Universal Installer (OUI) also uses OIFCFG to identify and display the interfaces available on the system.

Defining Network Interfaces with OIFCFG

The specification for a network interface uniquely identifies it using the interface name, its associated subnet, and interface type. The interface type indicates the purpose for which the network is configured. The supported interface types are:

- **Public**—An interface that can be used for communication with components external to RAC instances, such as Oracle Net and Virtual Internet Protocol (VIP) addresses
- **Cluster interconnect**—A private interface used for the cluster interconnect to provide inter-instance or Cache Fusion communication
- **Storage**—A disk interface used for high-speed file I/O

A network interface can be stored as a global interface or as a node-specific interface. An interface is stored as a global interface when all the nodes of a RAC cluster have the same interface connected to the same subnet (recommended). It is stored as a node-specific interface only when there are some nodes in the cluster that have a different set of interfaces and subnets. If an interface is configured as both a global and a node-specific interface, the node-specific definition takes precedence over the global definition.

A network interface specification is in the form of:

```
<interface_name>/<subnet>:<interface_type>.
```

For example, the following identifies qfe0 as a cluster interconnect located at the address 204.152.65.32:

```
qfe0/204.152.65.32:cluster_interconnect
```

Syntax and Commands for the OIFCFG Command-Line Tool

Use the `oifcfg -help` command to display online help for OIFCFG. The elements of OIFCFG commands, some of which are optional depending on the command, are:

- `nodename` Name of the host, as known to a communications network
- `if_name` Name by which the interface is configured in the system
- `subnet` Subnet address of the interface
- `if_type` Type of interface: `public`, `cluster_interconnect`, or `storage`

You can use OIFCFG to list the interface names and the subnets of all the interfaces available on the local node by executing the `iflist` keyword as shown in this example:

```
oifcfg iflist
hme0      139.185.141.0
qfe0      204.152.65.16
```

You can also retrieve specific OIFCFG information with a `getif` command using the following syntax:

```
oifcfg getif [ [-global | -node <nodename>] [-if <if_name>[/<subnet>]] [-type <if_type>] ]
```

To store a new interface use the `setif` keyword. For example, to store the interface `hme0`, with the subnet `139.185.141.0`, as a global interface (to be used as an interconnect for all the RAC instances in your cluster), you would use the command:

```
oifcfg setif -global hme0/139.185.141.0:cluster_interconnect
```

For a cluster interconnect that exists between only two nodes, for example rac1 and rac2, you could create the cms0 interface with the following commands, assuming 139.185.142.1 and 139.185.142.2 are the subnet addresses for the interconnect on rac1 and rac2 respectively:

```
oifcfg setif -node rac1 cms0/139.185.142.1:cluster_interconnect
oifcfg setif -node rac2 cms0/139.185.142.2:cluster_interconnect
```

Use the OIFCFG `delif` command to delete the stored configuration for global or node-specific interfaces. A specific node-specific or global interface can be deleted by supplying the interface name, with an optional subnet, on the command line. Without the `-node` or `-global` options, the `delif` keyword deletes either the given interface or all the global and node-specific interfaces on all the nodes in the cluster. For example, the following command deletes the global interface named qfe0 for the subnet 204.152.65.16:

```
oifcfg delif -global qfe0/204.152.65.16
```

On the other hand, the next command deletes all the global interfaces stored with OIFCFG:

```
oifcfg delif -global
```


A

Troubleshooting

This appendix explains how to use trace and log files in Oracle Real Application Clusters (RAC). This chapter also includes information about the alert file that is specific to each instance. The topics in this appendix are:

- [Monitoring Trace Files in Real Application Clusters](#)
- [Using Log Files in Real Application Clusters](#)
- [Enabling Additional Tracing for Real Application Clusters High Availability](#)
- [Using Instance-Specific Alert Files in Real Application Clusters](#)

Monitoring Trace Files in Real Application Clusters

Oracle records information about important events that occur in your RAC environment in trace files. The trace files for RAC are the same as those in single-instance Oracle databases. As a best practice, monitor trace files frequently and back them up regularly for all instances. Doing so preserves their content for future troubleshooting.

Where to Find Files for Analyzing Errors

Information about ORA-600 errors appear in the *alert_SID.log* file. For troubleshooting, you may need to also provide files from these bdump locations:

- `$ORACLE_HOME/admin/db_name/bdump` on UNIX-based systems
- `%ORACLE_HOME%\admin\db_name\bump` on Windows-based systems

Some files may also be in the `udump` directory.

Background Thread Trace Files in Real Application Clusters RAC background threads use trace files to record database operations and database errors. These trace logs help troubleshoot and also enable Oracle support to more efficiently debug cluster database configuration problems.

Background thread trace files are created regardless of whether the `BACKGROUND_DUMP_DEST` parameter is set in the server parameter file (SPFILE). If you set `BACKGROUND_DUMP_DEST`, then the trace files are stored in the directory specified. If you do not set the parameter, then the trace files are stored as follows:

- `$ORACLE_BASE/admin/db_name/bdump` on UNIX-based systems
- `%ORACLE_BASE%\admin\db_name\bump` on Windows-based systems

The Oracle database creates a different trace file for each background thread. On UNIX-based systems, the trace files for the background processes are named `SID_<process name>_<process identifier>.trc`, for example:

- `SID_dbwr_1202.trc`
- `SID_smon_4560.trc`

User Process Trace Files in Real Application Clusters Trace files are created for user processes if you set the `USER_DUMP_DEST` parameter in the initialization parameter file. The trace files for the user processes have the form `oraxxxxx.trc`, where `xxxxx` is a 5-digit number indicating the process identifier (PID) on UNIX-based systems or the thread number on Windows-based systems.

Using Log Files in Real Application Clusters

RAC provides several types of log files that record processing information as described in this section.

Clusterware Log Files

The following sections describe the locations of the clusterware log files.

Cluster Ready Services Log Files Cluster Ready Services (CRS) has daemon processes that generate log information. Log files for the CRS daemon (`crsd`) can be found in the following directories:


```
<CRS Home>/crs/init
<CRS Home>/crs/<node name>.log
```

Oracle Cluster Registry Log Files The Oracle Cluster Registry (OCR) records log information in the following location:

```
<CRS Home>/srvm/log/
```

Cluster Synchronization Services (CSS) Log Files You can find CSS information that the OCSSD generates in log files in the following locations:

```
<CRS Home>/css/log/ocssd<number>.log
<CRS Home>/css/init/<node_name>.log
```

Event Manager Log Files Event Manager (EVM) information generated by evmd is recorded in log files in the following locations:

```
<CRS Home>/evm/log/evmdaemon.log
<CRS Home>/evm/init/<node_name>.log
```

Oracle High Availability Log Files

The Oracle RAC high availability trace files are located in:

```
$ORACLE_BASE/<database_name>/admin/hdump
```

Where \$ORACLE_BASE is configured and \$ORACLE_HOME/racg/log when \$ORACLE_BASE is not available.

Enabling Additional Tracing for Real Application Clusters High Availability

You may be requested to enable tracing to capture additional information for problem resolution when working with Oracle Support. Because the procedures described in this section may adversely affect performance, only perform these activities with the assistance of Oracle Support.

Generating Additional Trace Information for a Running Resource To generate additional trace information for a running resource, set the resource attribute `_USR_ORA_DEBUG` to the value 1 using one of the following two methods:

- For an individual resource, modify the resource profile (a text file named `<resource_name>.cap`) that you can generate with the following command:

```
crs_stat -p <resource_name> > \
<CRS Home>/crs/profile/<resource_name>.cap
```

Only create cap files in this directory for node-level resources. Use the `<CRS Home>/crs/public/` directory for other resources. Then set the `_USR_ORA_DEBUG` value to 1 in this cap file and issue the following command:

```
crs_register -u <resource_name>
```

- For all resources, add the following line to the script, `racgwrap`, in `$ORACLE_HOME/bin`:

```
_USR_ORA_DEBUG=1
export _USR_ORA_DEBUG
```

Only node-level resources (VIP, Enterprise Manager, and so on) should have their cap file created in the directory <CRS Home>/crs/profile.

Verifying Event Manager Daemon Communications The event manager daemons (evmd) running on separate nodes communicate through specific ports. To determine whether the evmd for a node can send and receive messages, perform the test described in this section while running session 1 in the background.

On node 1, session 1 enter:

```
$ evmwatch -A -t "@timestamp @@"
```

On node 2, session 2 enter:

```
$ evmpost -u "hello" [-h nodename]
```

Session 1 should show output similar to the following:

```
$ 21-Aug-2002 08:04:26 hello
```

Ensure that each node can both send and receive messages by executing this test in several permutations.

Enabling Additional Debugging Information for Cluster Ready Services The `crsd` daemon can produce additional debugging information if you set the variable `CRS_DEBUG` to the value 1 by performing the following procedures:

In the file `/etc/init.d/init.crsd` add the entry:

```
CRS_DEBUG=1
export CRS_DEBUG
```

Then kill the `crsd` daemon with the command:

```
$ kill -9 <crsd process id>
```

Allow the `init` process to restart the `crsd`. Oracle will write additional information to the standard log files.

Enabling Tracing for Java-Based Tools and Utilities in Real Application Clusters All of the Java-based tools and utilities available in RAC are invoked by executing scripts of the same name as the tool or utility. This includes the Database Configuration Assistant (DBCA), the Net Configuration Assistant (NetCA), the Virtual Internet Protocol Configuration Assistant (VIPCA), Service Control (SRVCTL), and the Global Services Daemon (GSD). For example to run the DBCA, enter the command `dbca`.

To enable additional debugging information, edit the command file (such as `dbca` in this example) and add the following parameters to the `$JRE_DIR/bin/jre` command line:

```
-DTRACING.ENABLED=true -DTRACING.LEVEL=2
```

For example, the file `$ORACLE_HOME/bin/dbca` contains the line:

```
$JRE_DIR/bin/jre -DORACLE_HOME=$OH -DJDBC_PROTOCOL=thin -mx64 -classpath
$CLASSPATH oracle.sysman.dbca.Dbca $ARGUMENTS
```

Change this line to:

```
$JRE_DIR/bin/jre -DORACLE_HOME=$OH -DTRACING.ENABLED=true -DTRACING.LEVEL=2
-DJDBC_PROTOCOL=thin -mx64 -classpath $CLASSPATH oracle.sysman.dbca.Dbca
$ARGUMENTS
```

When you run the DBCA, trace information will record in the DBCA log file.

To debug SRVCTL, add the argument `-D` to the command line. For example, to generate tracing information for an add database operation, execute the following command:

```
$ srvctl -D add database -d mydatabase
```

Using Instance-Specific Alert Files in Real Application Clusters

Each instance in the cluster database has one alert file. The alert file for each instance, *alert_SID.log*, contains important information about error messages and exceptions that occur during database operations. Information is appended to the alert file each time you start the instance. All process threads can write to the alert file for the instance.

The *alert_SID.log* file is in the directory specified by the `BACKGROUND_DUMP_DEST` parameter in the *initdb_name.ora* initialization parameter file. If you do not set the `BACKGROUND_DUMP_DEST` parameter, the *alert_SID.log* file is generated in the following locations:

- `$ORACLE_BASE/admin/db_name/bdump` on UNIX-based systems.
- `%ORACLE_BASE%\admin\db_name\bump` on Windows-based system

Resolving Pending Shutdown Issues

In some situations a `SHUTDOWN IMMEDIATE` may be pending and Oracle will not quickly respond to repeated shutdown requests. This is because Cluster Ready Services (CRS) may be processing a current shutdown request. In such cases, issue a `SHUTDOWN ABORT` using SQL*Plus for subsequent shutdown requests.

Server Control (SRVCTL) Reference

This chapter describes how to administer Oracle Real Application Clusters (RAC) databases and instances using the Server Control (SRVCTL) utility. The topics in this chapter include:

- [Overview of SRVCTL for Administering Real Application Clusters](#)
- [SRVCTL Command Syntax and Options](#)
- [SRVCTL General Cluster Database Administration Tasks](#)
- [SRVCTL Cluster Database Configuration Tasks](#)
- [SRVCTL Node-Level Tasks](#)
- [SRVCTL Command Reference](#)
- [SRVCTL Commands](#)

See Also: *Oracle Real Application Clusters Installation and Configuration Guide* for information about using the DBCA

Overview of SRVCTL for Administering Real Application Clusters

The Server Control (SRVCTL) utility is installed on each node by default. You can use SRVCTL to start and stop the database and instances, manage configuration information, and to delete or move instances and services. SRVCTL also manages configuration information.

Some SRVCTL operations store configuration information in the Oracle Cluster Registry (OCR). SRVCTL performs other operations, such as starting and stopping instances, by sending requests to the Cluster Ready Services daemon (CRSD), which then starts or stops the Cluster Ready Services (CRS) resources.

To use SRVCTL, enter the `srvctl` command and its options in case sensitive syntax as described under the heading "[SRVCTL Command Reference](#)" on page B-4.

Guidelines for Using SRVCTL in Real Application Clusters

Guidelines for using SRVCTL are:

- To use SRVCTL to change your RAC database configuration, log in to the database as the `oracle` user. Members of the DBA group can start and stop the database.
- Only use the version of SRVCTL that is provided with Oracle Database 10g on RAC databases that are created or upgraded for Oracle Database 10g.
- Always use SRVCTL from the `ORACLE_HOME` of the database that you are administering.
- SRVCTL does not support concurrent executions of commands on the same object. Therefore, only run one SRVCTL command at a time for each database, service, or other object.

Obtaining Command Line Help for SRVCTL

To see help for all SRVCTL commands, from the command line enter:

```
srvctl -h
```

To see the command syntax and a list of options for each SRVCTL command, from the command line enter:

```
srvctl <command> (or verb) <object> (or noun) -h
```

To see the SRVCTL version number enter:

```
srvctl -V
```

Caution: Although you may be able to cancel running SRVCTL commands by entering Control-C at the command line, you may corrupt your configuration data by doing this. You are strongly advised not to attempt to terminate SRVCTL in this manner.

SRVCTL Command Syntax and Options

SRVCTL commands, objects, and options are case sensitive. Database, instance, and service names are case insensitive and case preserving. SRVCTL interprets the following command syntax:

```
srvctl command object [options]
```

In SRVCTL syntax:

- `srvctl` is the command to start the SRVCTL utility.
- *command* is a verb such as `start`, `stop`, or `remove`.
- *object* is an object or target on which SRVCTL performs the command, such as *database* or *instance*. You can also use object abbreviations.
- *options* extend the use of a preceding command combination to include additional parameters for the command. For example, the `-i` option indicates that a comma-delimited list of instance names follows; sometimes the `-i` option only permits one value and not a list of names. The `-n` option indicates that a node name or a comma-delimited list of node names follows. SRVCTL prompts for user credentials when you use the `-q` option with any SRVCTL command.

SRVCTL Cluster Database Configuration Tasks

The database configuration tasks are:

- Add, modify, and delete cluster database configuration information.
- Add an instance or a service to, and delete an instance or service from the configuration of a cluster database.
- Move instances and services in a cluster database configuration and modify service configurations.
- Set and unset the environment for an instance or service in a cluster database configuration.
- Set and unset the environment for an entire cluster database in a cluster database configuration.

SRVCTL General Cluster Database Administration Tasks

The general database administration tasks are:

- Start and stop cluster databases
- Start and stop cluster database instances
- Start, stop, and relocate cluster database services
- Obtain statuses of cluster databases, cluster database instances, or cluster database services

SRVCTL Node-Level Tasks

The node-level tasks are:

- Adding and deleting node level applications.
- Setting and unsetting the environment for node-level applications.
- Administering node applications.
- Administering ASM instances.
- Starting and stopping a group of programs that includes virtual IP addresses, listeners, Oracle Notification Services, and Oracle Enterprise Manager agents (for maintenance purposes).

SRVCTL Command Reference

This section presents the SRVCTL command reference:

- [srvctl add](#)
- [srvctl config](#)
- [srvctl enable](#)
- [srvctl disable](#)
- [srvctl start](#)
- [srvctl stop](#)
- [srvctl modify](#)
- [srvctl relocate](#)
- [srvctl status](#)
- [srvctl getenv](#)
- [srvctl setenv and unsetenv](#)
- [srvctl remove](#)

SRVCTL Commands

This section summarizes the SRVCTL commands, objects, and options. Oracle recommends that you use the DBCA to create your RAC database as well as your initial service configurations. This is because the DBCA configures both the CRS resources and the Net Service entries for each service.

SRVCTL Commands Summary

[Table B-1](#) summary the SRVCTL commands. Execute SRVCTL commands from the command line and specify one or more objects with the appropriate options for the command and its objects.

Table B-1 SRVCTL Commands Summary

Command	Description
srvctl add on page B-5	Adds the node applications, database, database instance, ASM instance, or service.
srvctl remove on page B-8	Removes the node applications, database, database instance, ASM instance, or service.
srvctl config on page B-8	Lists the configuration for the node applications, database, ASM instance, or service.
srvctl enable on page B-10	Enables the database, database instance, ASM instance, or service.
srvctl disable on page B-12	Disables the database, database instance, ASM instance, or service.
srvctl start on page B-14	Starts the node applications, database, database instance, ASM instance, or service.
srvctl stop on page B-17	Stops the node applications, database, database instance, ASM instance, or service.
srvctl modify on page B-17	Modifies the node applications, database, database instance, or service configuration.

Table B–1 (Cont.) SRVCTL Commands Summary

Command	Description
srvctl relocate on page B-23	Relocates the service from one instance to another.
srvctl status on page B-24	Obtains the status of the node applications, database, database instance, ASM instance, or service.
srvctl getenv on page B-26	Displays the environment variable in the configuration for the node applications, database, database instance, or service.
srvctl setenv and unsetenv on page B-28	Sets and unsets the environment variable in the configuration for the node applications, database, database instance, or service.

SRVCTL Objects Summary

[Table B–2](#) lists the SRVCTL objects for SRVCTL commands. Use the full name or the abbreviation for the purpose described.

Table B–2 SRVCTL Objects (Nouns) and Abbreviations

Object Noun Name	Abbreviation	Purpose
asm	asm	To add, configure, enable, start, obtain the status of, stop, disable, and remove ASM instances.
database	db	To add, configure, modify, manage environment variables for, enable, start, obtain the status of, stop, disable, and remove databases.
instance	inst	To add, configure, modify, manage environment variables for, enable, start, obtain the status of, stop, and remove database instances.
nodeapps	no abbreviation	To add, configure, modify, manage environment variables for, start, obtain the status of, stop, and remove node applications.
service	serv	To add, configure, modify, manage environment variables for, enable, start, obtain the status of, relocate, disable, stop, and remove services from your cluster database.

srvctl add

The SRVCTL add command adds the configuration and the CRS applications to the OCR for the cluster database, named instances, named services, or for the named nodes. To execute `srvctl add` operations, you must be logged in as the database administrator and be the Oracle account owner on UNIX-based systems, or you must be logged on as a user with Administrator privileges on Windows-based systems.

When adding an instance, the name that you specify with `-i` must match the `ORACLE_SID` parameter. The database name given with `-d db_unique_name` must match the `DB_UNIQUE_NAME` initialization parameter setting. If `DB_UNIQUE_NAME` is unspecified, then match the `DB_NAME` initialization parameter setting. The default setting for `DB_UNIQUE_NAME` uses the setting for `DB_NAME`. Also, the domain name given with `-m db_domain` must match the `DB_DOMAIN` setting.

Table B–3 srvctl add Summary

Command	Description
srvctl add database on page B-6	Adds a database and configuration.
srvctl add instance on page B-6	Adds one or more instances and configurations.
srvctl add service on page B-7	Adds services.

Table B-3 (Cont.) *srvctl add* Summary

Command	Description
srvctl add nodeapps on page B-8	Adds node applications.
srvctl add asm on page B-8	Adds ASM instances.

srvctl add database

Adds a database configuration to your cluster database configuration.

Syntax and Options Use the `srvctl add database` command with the following syntax:

```
srvctl add database -d <db_unique_name> -o <oracle_home> [-m <domain_name>] [-p
<spfile>] [-A <addr_str>] [-r {PRIMARY |
PHYSICAL_STANDBY | LOGICAL_STANDBY}] [-s <start_options>] [-n <db_name>]
```

Table B-4 *srvctl add database* Options

Syntax	Description
-d <db_unique_name>	Unique name for the database.
-o <oracle_home>	The Oracle home for the database.
-m <domain_name>	The Domain for the database.
-p <spfile>	The server parameter file for the database.
-A <addr_str>	The database cluster alias (<name ip>/netmask[/if1[if2 ...]]).
-r {PRIMARY PHYSICAL_STANDBY LOGICAL_STANDBY}	The role of the database (primary, physical standby, or logical standby).
-s <start_options>	Startup options for the database.
-n <db_name>	The name of the database, where it is different from the unique name given by the -d option.

Example An example of this command is:

```
srvctl add database -d crm -o /ora/ora10
```

srvctl add instance

Adds a configuration for an instance to your cluster database configuration.

Syntax and Options Use the `srvctl add instance` command with the following syntax:

```
srvctl add instance -d <db_unique_name> -i <inst_name> -n <node_name>
```

Table B-5 *srvctl add instance* Options

Option	Description
-d <db_unique_name>	Unique name for the database.
-i <inst_name>	The instance name.
-n <node_name>	The node name.

Examples Examples of this command are:

```
srvctl add instance -d crm -i crm01 -n gm01
srvctl add instance -d crm -i crm02 -n gm02
srvctl add instance -d crm -i crm03 -n gm03
```

srvctl add service

Adds services to a database and assigns them to instances. If you have multiple instances of a cluster database on the same node, then always use only one instance on that node for all of the services that node manages. Also, you can use the `srvctl add service` command to configure the Transparent Application Failover (TAF) policy for a service.

See Also: [Chapter 4](#) for more information about TAF policies

Syntax and Options Use the `srvctl add service` command with the following syntax:

```
srvctl add service -d <db_unique_name> -s <service_name> -r <preferred_list>
[-a <available_list>] [-P <TAF_policy>]
```

Table B-6 *srvctl add service Options*

Option	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	The service name.
-r <preferred_list>	The list of preferred instances.
-a <available_list>	The list of available instances
-P <TAF_policy>	The TAF policy (NONE, BASIC, or PRECONNECT).

Use the following syntax to add a new preferred or available instance to the service configuration:

```
srvctl add service -d <db_unique_name> -s <service_name>
-u [-r <new_preferred_inst> | -a <new_available_inst>]
```

Table B-7 *srvctl add service Options for a New Instance*

Option	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	The service name.
-u	Add a new instance to service configuration
-r <new_preferred_inst>	Name of the new preferred instance.
-a <new_available_inst>	Name of new available instance.

Examples Use this example syntax to add a named service to a database with preferred instances in list one and available instances in list two, using basic failover for the available instances:

```
srvctl add service -d crm -s sales -r crm01,crm02 -a crm03
```

Use this example syntax to add a named service to a database with preferred instances in list one and available instances in list two, using preconnect failover for the available instances:

```
srvctl add service -d crm -s sales -r crm01,crm02 -a crm03 -P Preconnect
```

srvctl add nodeapps

Adds a node application configuration to the specified node.

Note: On UNIX-based systems, you must be logged in as `root` and on Windows-based systems, you must be logged in as a user with Administrator privileges to execute this command.

Syntax and Options Use the `srvctl add nodeapps` command with the following syntax:

```
srvctl add nodeapps -n <node_name> -o <oracle_home> -A <addr_str>
```

Table B–8 *srvctl add nodeapps Options*

Option	Description
-n <node_name>	Node name.
-o <oracle_home>	Oracle home for the cluster database.
-A <addr_str>	The node level VIP address (<name ip>/netmask[/if1[if2 ...]]).

Example An example of this is:

```
srvctl add nodeapps -n crmnode1 -o /ora/ora10 -A 1.2.3.4/255.255.255.0
```

srvctl add asm

Adds a record for an ASM instance to the specified node.

Syntax and Options Use the `srvctl add asm` command with the following syntax:

```
srvctl add asm -n <node_name> -i <asm_instance_name> -o <oracle_home>
```

Table B–9 *srvctl add asm Options*

Option	Description
-n <node_name>	Node name.
-i <asm_instance_name>	The ASM instance name.
-o <oracle_home>	Oracle home for the cluster database.

Example An example of this is:

```
srvctl add asm -n crmnode1 -i asm1 -o /ora/ora10
```

srvctl config

The `SRVCTL config` command displays the configuration stored in the OCR.

Table B–10 *srvctl config Summary*

Command	Description
srvctl config database on page B-9	Displays the configuration information of the cluster database.
srvctl config service on page B-9	Displays the configuration information for the services.
srvctl config nodeapps on page B-10	Displays the configuration information for the node applications.
srvctl config asm on page B-10	Displays the configuration for the ASM instances on the node.

srvctl config database

Displays the configuration for a RAC database or lists all configured databases.

Syntax and Options Use the `srvctl config database` command with the following syntax:

```
srvctl config database [-d <db_unique_name> [-a] [-t]]
```

Table B–11 *srvctl config database Options*

Option	Description
-d <db_unique_name>	Unique name for the database
-a	Additional attributes
-t	Display sample TNS entries

Examples An example of this command to list all database is:

```
srvctl config database
```

An example of this command to show sample TNS entries for a specific database is:

```
srvctl config database -d db_erp
```

srvctl config service

Displays the configuration for a service.

Syntax and Options Use the `srvctl config service` command with the following syntax:

```
srvctl config service -d <db_unique_name> [-s <service_name>] [-a]
```

Table B–12 *srvctl config service Options*

Option	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	Service name.
-a	Additional attributes.

Example An example of this command is:

```
srvctl config service -d crm -s crm
```

Note: If you do not specify `-s <service>`, then SRVCTL displays all services for the specified cluster database.

srvctl config nodeapps

Displays the configuration for node applications.

Syntax and Options Use the `srvctl config nodeapps` command with the following syntax:

```
srvctl config nodeapps -n <node_name> [-a] [-g] [-s] [-l]
```

Table B-13 *srvctl config nodeapps Option*

Option	Description
-n <node_name>	Node name.
-a	VIP configuration.
-g	GSD configuration.
-s	ONS configuration.
-l	Listener configuration.

Example An example of this is:

```
srvctl config nodeapps -n mynode1
```

srvctl config asm

Displays the configuration for the ASM instances.

Syntax and Options Use the `srvctl config asm` command with the following syntax:

```
srvctl config asm -n <node_name>
```

The only option available for this command is `-n` to specify the node name.

Example An example of this is:

```
srvctl config asm -n mynode1
```

srvctl enable

The SRVCTL `enable` command enables the named object so that it can run under CRS for automatic startup, failover, or restart. The CRS application supporting the object may be up or down to use this function. `enable` is the default value. If the object is already enabled, then the command is ignored. Enabled objects can be started, and disabled objects cannot be started.

Table B-14 *srvctl enable Summary*

Command	Description
srvctl enable database on page B-11	Enables the database.
srvctl enable instance on page B-11	Enables the instance.

Table B-14 (Cont.) *srvctl enable* Summary

Command	Description
srvctl enable service on page B-11	Enables a service.
srvctl enable asm on page B-12	Enables an ASM instance.

srvctl enable database

Enables CRS resources for a database and enables the database's instances if the database was previously disabled.

Syntax and Options Use the `srvctl enable database` command with the following syntax:

```
srvctl enable database -d <db_unique_name>
```

The only option available for this command is `-d` to specify the database name.

Example An example of this command is:

```
srvctl enable database -d crm
```

srvctl enable instance

Enables an instance for CRS. If all instances are disabled, then enabling an instance also enables the database.

Syntax and Options Use the `srvctl enable instance` command with the following syntax:

```
srvctl enable instance -d <db_unique_name> -i <inst_name_list>
```

Table B-15 *srvctl enable instance* Option

Option	Description
<code>-d <db_unique_name></code>	Unique name for the database
<code>-i <inst_name_list></code>	Comma-separated list of instance names.

Example An example of this command is:

```
srvctl enable instance -d crm -i crm1,crm2...
```

srvctl enable service

Enables a service for CRS. Enabling an entire service also affects the enabling of the service over all the instances by enabling the service at each one. When the entire service is already enabled, an `srvctl enable service` operation does not affect all the instances and enable them. Instead, this operation returns an error. Therefore, you cannot always use the entire set of service operations to manipulate the per-instance service indicators.

Syntax and Options Use the `srvctl enable service` command with the following syntax:

```
srvctl enable service -d <db_unique_name> {-s <service_name_list> | -s <service_name> -i <inst_name>}
```

Table B–16 *srvctl enable service Options*

Option	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name_list>	Comma-separated list of service names.
-s <service_name>	Single service name.
-i <inst_name>	Instance name.

Examples The following example globally enables a service:

```
srvctl enable service -d crm -s crm
```

The following example enables a service to use a preferred instance:

```
srvctl enable service -d crm -s crm -i crml
```

srvctl enable asm

Enables an ASM instance.

Syntax and Options Use the `srvctl enable asm` command with the following syntax:

```
srvctl enable asm -n <node_name> [-i <asn_inst_name>]
```

Table B–17 *srvctl enable asm Option*

Option	Description
-n <node_name>	Node name
-i <inst_name>	ASM instance name.

Example An example of this command is:

```
srvctl enable asm -n crmnode1 -i asml
```

srvctl disable

Disables a specified object (cluster database, database instance, ASM instance, or service). `SRVCTL disable` is intended to be used when an object is to be repaired or is down for maintenance to prevent inappropriate automatic restarts. When you issue the `disable` command, the object is disabled and unavailable to run under CRS for automatic startup, failover, or restart. If you specify `-i <instance_name>`, then `SRVCTL` only disables the service from running on the specified instance.

Table B–18 *srvctl disable Summary*

Command	Description
srvctl disable database on page B-13	Disables the cluster database
srvctl disable instance on page B-13	Disables an instance
srvctl disable service on page B-13	Disables a service
srvctl disable asm on page B-14	Disables an ASM instance

srvctl disable database

Disables a cluster database and its instances.

Syntax and Options Use the `srvctl disable database` command with the following syntax:

```
srvctl disable database -d <db_unique_name>
```

The only option available for this command is `-d` to specify the database name.

Example An example of this command is:

```
srvctl disable database -d mybd1
```

srvctl disable instance

Disables an instance. If the instance that you disable with this command is the last enabled instance, then this operation also disables the database.

Syntax and Options Use the `srvctl disable instance` command with the following syntax:

```
srvctl disable instance -d <db_unique_name> -i <inst_name_list>
```

Table B–19 *srvctl disable instance Options*

Option	Description
<code>-d <db_unique_name></code>	Unique name for the database.
<code>-i <inst_name_list></code>	Comma-separated instance names.

Example An example of this command is:

```
srvctl disable instance -d crm -i crm1,crm3
```

srvctl disable service

Disables a service. Disabling an entire service affects all the instances, disabling each one. When the entire service is already disabled, an `srvctl disable service` operation on the entire service affect all the instances and disable them; it just returns an error. This means that you cannot always use the entire set of service operations to manipulate the per-instance service indicators.

Syntax and Options Use the `srvctl disable service` command with the following syntax:

```
srvctl disable service -d <db_unique_name> {-s <service_name_list> | -s <service_name>} -i <inst_name>}
```

Table B–20 *srvctl disable service Options*

Option	Description
<code>-d <db_unique_name></code>	Unique name for the database.
<code>-s <service_name_list></code>	Comma-separated service names.

Table B–20 (Cont.) *srvctl disable service Options*

Option	Description
-s <service_name>	Single service name.
-i <inst_name>	Instance name.

Examples The following example globally disables a service:

```
srvctl disable service -d crm -s crm,marketing
```

The following example disables a service running on the preferred named instance and results in running a service on one less instance:

```
srvctl disable service -d crm -s crm -i crml
```

srvctl disable asm

Disables an ASM instance.

Syntax and Options Use the `srvctl disable asm` command with the following syntax:

```
srvctl disable asm -n <node_name> [-i <asn_inst_name>]
```

Table B–21 *srvctl disable asm Option*

Option	Description
-n <node_name>	Node name
-i <inst_name>	ASM instance name.

Example An example of this command is:

```
srvctl disable asm -n crmnode1 -i asml
```

srvctl start

Starts CRS enabled, non-running applications for the database, all or named instances, all or named service names, or node-level applications. For the `start` command, and for other operations that use a connect string, if you do not provide a connect string, then SRVCTL uses `"/ as sysdba"` to perform the operation. To run such operations, the owner of the `oracle` binary executables must be a member of the OSDBA group, and users running the commands must be in the OSDBA group also.

Table B–22 *srvctl start Summary*

Command	Description
srvctl start database on page B-15	Starts the cluster database and its instances
srvctl start instance on page B-15	Starts the instance
srvctl start service on page B-15	Starts the service
srvctl start nodeapps on page B-16	Starts the node applications
srvctl start asm on page B-16	Starts ASM instances

srvctl start database

Starts a cluster database and its enabled instances.

Syntax and Options Use the `srvctl start database` command with the following syntax:

```
srvctl start database -d <db_unique_name> [-o <start_options>] [-c <connect_str> | -q]
```

Table B–23 *srvctl start database Options*

Option	Description
-d <db_unique_name>	Unique name for the database.
-o <start_options>	Options for startup command (for example: open, mount, or nomount)
-c <connect_str>	Connect string (default: / as sysdba)
-q	Prompt for user credentials connect string from standard input.

Example An example of this command is:

```
srvctl start database -d crm -o open
```

srvctl start instance

Starts instances in the cluster database.

Syntax and Options Use the `srvctl start instance` command with the following syntax:

```
srvctl start instance -d <db_unique_name> -i <inst_name_list> [-o <start_options>] [-c <connect_str> | -q]
```

Table B–24 *srvctl start instance Options*

Option	Description
-d <db_unique_name>	Unique name for the database.
-i <inst_name_list>	Comma-separated instance names.
-o <start_options>	Options for startup command (for example: open, mount, or nomount).
-c <connect_str>	Connect string (default: / as sysdba).
-q	Prompt for user credentials connect string from standard input.

Example An example of this command is:

```
srvctl start instance -d crm -i crm1,crm4
```

srvctl start service

Starts a service or multiple services on the specified instance. The `srvctl start service` command will fail if you attempt to start a service on an instance if that service is already running on its maximum number of instances, that is, its number of preferred instances. You may move a service or change the status of a service on an

instance with the [srvctl modify service](#) and [srvctl relocate service](#) commands described on page B-21 and on page B-24 respectively.

Syntax and Options Use the `srvctl start service` command with the following syntax:

```
srvctl start service -d <db_unique_name> [-s <service_name_list> [-i <inst_name>]]
[-o <start_options>] [-c <connect_str> | -q]
```

Table B-25 *srvctl start service Options*

Option	Description
-d <db_unique_name>	Unique name for the database
-s <service_name_list>	Comma-separated service names; the service name list is optional and if not provided, the SRVCTL starts all of the database's services
-i <inst_name>	Instance name
-o <start_options>	Options to startup command (for example: open, mount, or nomount)
-c <connect_str>	Connect string (default: / as sysdba)
-q	Query connect string from standard input

Examples The following example starts named service names. If the instances that support these services, including available instances that the service uses for failover, are not running but are enabled, then they are started:

```
srvctl start service -d crm -s crm
```

The following example starts a named service on a specified instance:

```
srvctl start service -d crm -s crm -i crm2
```

srvctl start nodeapps

Starts node-level applications on a particular node.

Syntax and Options Use the `srvctl start nodeapps` command with the following syntax:

```
srvctl start nodeapps -n <node_name>
```

The only option available for this command is `-n` to specify the node name.

Example An example of this command is:

```
srvctl start nodeapps -n mynode1
```

srvctl start asm

Starts an ASM instance.

Syntax and Options Use the `srvctl start asm` command with the following syntax:

```
srvctl start asm -n <node_name> [-i <asn_inst_name>] [-o <start_options>]
```

Table B–26 *srvctl start asm Option*

Option	Description
-n <node_name>	Node name
-i <inst_name>	ASM instance name.
-o <start_options>	Options to startup command (for example: open, mount, or nomount)

Examples An example of this command to start a single ASM instance is:

```
srvctl start asm -n crmnode1 -i asm1
```

An example to start all ASM instances on a node is:

```
srvctl start asm -n crmnode2
```

srvctl stop

Stops the CRS applications for the database, all or named instances, all or named service names, or node level applications. Only CRS applications that are starting or running are stopped. Objects running outside of CRS are not stopped. Stops node-level applications and all dependent CRS applications on the node.

You should disable an object that you intend to remain stopped after you issue a SRVCTL stop command. Refer to the SRVCTL `disable` command starting with [srvctl disable database](#) on page B-13.

Note: If the object is stopped and is not disabled, then it can restart as a result of another planned operation. That is, the object will *not* restart as a result of a failure. Oracle recommends that you disable an object that should remain stopped after you issue a `stop` command.

Table B–27 *srvctl stop Summary*

Command	Description
srvctl stop database on page B-17	Stops the cluster database
srvctl stop instance on page B-18	Stops the instance
srvctl stop service on page B-18	Stops the service
srvctl stop nodeapps on page B-19	Stops the node-level applications
srvctl stop asm on page B-19	Stops ASM instances

srvctl stop database

Stops a database, its instances, and its services.

Syntax and Options Use the `srvctl stop database` command with the following syntax:

```
srvctl stop database -d <db_unique_name> [-o <stop_options>] [-c <connect_str> | -q]
```

Table B–28 *srvctl stop database Options*

Option	Description
-d <db_unique_name>	Unique name for the database
-o <stop_options>	shutdown command options (for example: normal, transactional, immediate, or abort)
-c <connect_str>	Connect string (default: / as sysdba)
-q	Prompt for user credentials connect string from standard input

Example An example of this command is:

```
srvctl stop database -d crm
```

srvctl stop instance

Stops instances and stops all enabled and non-running services that have these instances as either preferred or available instances.

Syntax and Options Use the `srvctl stop instance` command with the following syntax:

```
srvctl stop instance -d <db_unique_name> -i <inst_name_list> [-o <stop_options>]
[-c <connect_str> | -q]
```

Table B–29 *srvctl stop instance Options*

Option	Description
-d <db_unique_name>	Unique name for the database
-i <inst_name_list>	Comma-separated instance names
-o <stop_options>	Options for shutdown command (for example: normal, transactional, immediate, or abort)
-c <connect_str>	Connect string (default: / as sysdba)
-q	Query connect string from standard input

Example An example of this command is:

```
srvctl stop instance -d crm -i crml
```

srvctl stop service

Stops one or more services globally across the cluster database, or on the specified instance.

Syntax and Options Use the `srvctl stop service` command with the following syntax:

```
srvctl stop service -d <db_unique_name> [-s <service_name_list> [-i <inst_name>]]
[-c <connect_str> | -q] [-f]
```

Table B–30 *srvctl stop service Options*

Option	Description
-d <db_unique_name>	Unique name for the database
-s <service_name_list>	Comma-separated service names; if you do not provide a service name list, then SRVCTL stops all services on the database
-i <inst>	Instance name
-c <connect_str>	Connect string (default: / as sysdba)
-q	Query connect string from standard input
-f <force>	Force SRVCTL to stop the service; this causes SRVCTL to disconnect all the sessions transactionally, causing the sessions using the service to reconnect to another instance

Examples The following example stops a service globally across a cluster database:

```
srvctl stop service -d crm -s crm
```

The following example stops a service on a specified instance:

```
srvctl stop service -d crm -s crm -i crm2
```

srvctl stop nodeapps

Stops node-level applications on a particular node.

Syntax and Options Use the `srvctl stop nodeapps` command with the following syntax:

```
srvctl stop nodeapps -n <node_name>
```

The only option for this command is the `-n` option, which specifies the node name.

Example An example of this command is:

```
srvctl stop nodeapps -n mynode1
```

srvctl stop asm

Stops an ASM instance.

Syntax and Options Use the `srvctl stop asm` command with the following syntax:

```
srvctl stop asm -n <node_name> [-i <inst_name>] [-o <stop_options>]
```

Table B–31 *srvctl stop asm Option*

Option	Description
-n <node_name>	Node name
-i <inst_name>	ASM instance name.
-o <stop_options>	Options for shutdown command (for example: normal, transactional, immediate, or abort)

Example An example of this command is:

```
srvctl stop asm -n crmnode1 -i asml
```

srvctl modify

Enables you to modify the instance configuration without removing and adding the CRS resources. Using `modify` preserves the environment in the OCR configuration that would otherwise need to be re-entered. The configuration description is modified in the OCR configuration, and a new CRS profile is generated and registered. The change takes effect when the application is next restarted.

Table B–32 *srvctl modify Summary*

Command	Description
srvctl modify database on page B-20	Modifies the configuration for a database.
srvctl modify instance on page B-20	Modifies the configuration for an instance.
srvctl modify service on page B-21	Modifies the configuration for a service.
srvctl modify nodeapps on page B-23	Modifies the configuration for a node application.

srvctl modify database

Modifies the configuration for a database.

Syntax and Options Use the `srvctl modify database` command with the following syntax:

```
srvctl modify database -d <db_unique_name> [-n <db_name>] [-o <ohome>] [-m
<domain_name>] [-p <spfile>] [-r {PRIMARY | PHYSICAL_STANDBY | LOGICAL_STANDBY}]
[-s <start_options>]
```

Table B–33 *srvctl modify database Options*

Option	Description
-d <db_unique_name>	Unique name for the database.
-n <db_name>	Name of the database, where it is different from the unique name given by -d option.
-o <ohome>	Oracle home for cluster database.
-m <domain_name>	Domain for cluster database.
-p <spfile>	Server parameter file for cluster database.
-r <role> [PRIMARY PHYSICAL_STANDBY LOGICAL_STANDBY]	Role of the database (primary, physical standby, or logical standby)
-s <start_options>	Startup options for the database

Example The following example changes the role of a database to a logical standby:

```
srvctl modify database -d crm -r logical_standby
```

srvctl modify instance

Modifies the configuration for a database instance from its current node to another node or changes the dependency between and ASM instance and a database instance.

Syntax and Options Use the `srvctl modify instance` command with the following syntax:

```
srvctl modify instance -d <db_unique_name> -i <inst_name> {-n <node_name> | -s
<asm_instance_name> | -r}
```

Table B-34 *srvctl modify instance Options*

Option	Description
-d <db_unique_name>	Unique name for the database.
-i <inst_name>	Database instance name.
-n <node_name>	Node name.
-s <asm_instance_name>	ASM instance dependency to database instance.
-r	Remove ASM instance dependency from database instance.

Examples An example of this command to relocate a database instance is:

```
srvctl modify instance -d crm -i crml -n my_new_node
```

The following example of this command establishes a dependency between an ASM instance and a database instance:

```
srvctl modify instance -d crm -i crml -s asml
```

srvctl modify service

Moves a service member from one instance to another. Additionally, this command changes which instances are to be the preferred and the available instances for a service. This command supports some online modifications to the service, for example:

- When a service configuration is modified so that a new preferred or available instance is added, the running state of the existing service is not affected. However, the newly added instances will not automatically provide the service, until a [srvctl start service](#) command is issued as described on page B-15.
- When there are available instances for the service, and the service configuration is modified so that a preferred or available instance is removed, the running state of the service may change unpredictably:
 - The service is stopped and then removed on some instances according to the new service configuration.
 - The service may be running on some instances that are being removed from the service configuration.
 - These services will be relocated to the next "free" instance in the new service configuration.

As a result of these considerations, when the online service is being modified, users may experience a brief service outage on some instances even if the instances are not being removed. Or users may experience a brief service outage on instances that are being removed from the service.

Important: Oracle recommends that you limit configuration changes to the minimum requirement and that you not perform other service operations while the online service modification is in progress.

Syntax and Options Use the `srvctl modify service` command with the following syntax:

```
srvctl modify service -d <db_unique_name> -s <service_name> -i <old_inst_name>
-t <new_inst_name> [-f]
```

Table B–35 *srvctl modify service Options for an Available Instance*

Option	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	Service name.
-i <old_inst_name>	Old instance name.
-t <new_inst_name>	New instance name.
-f	Disconnect all sessions during stop or relocate service operations.

You can also use the `srvctl modify service` command to change an available instance to a preferred instance as follows:

```
srvctl modify service -d <db_unique_name> -s <service_name> -i <avail_inst_name>
-r [-f]
```

Table B–36 *srvctl modify service Options for Changing an Available Instance to Preferred*

Option	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	Service name.
-i <avail_inst_name>	Instance name.
-r	Upgrade instance to preferred.
-f	Disconnect all sessions during stop or relocate service operations.

Examples An example of moving a service member from one instance to another is:

```
srvctl modify service -d crm -s crm -i crm1 -t crm2
```

An example of changing an available instance to a preferred instance is:

```
srvctl modify service -d crm -s crm -i crm1 -r
```

To change the status of multiple instances, you can use the `srvctl modify service` command to list which instances are to be the preferred and which are to be the available instances for a service as follows:

```
srvctl modify service -d <db_unique_name> -s <service_name> -n -i <pref_inst_list>
[-a <avail_inst_list>] [-f]
```

Table B–37 *srvctl modify service Options for Changing Instances between Preferred and Available status*

Option	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	Service name.
-n	Uses only the instances named for this service (unnamed instances already assigned to the service are removed).
-i <avail_inst_name>	List of preferred instances.
-a <avail_inst_list>	List of available instances.
-f	Disconnect all sessions during stop or relocate service operations.

Example An example of this command to exchange a preferred and available instance is:

```
srvctl modify service -d crm -s crm -n -i crm1 -a crm2
```

srvctl modify nodeapps

Applies a new Oracle home or virtual IP address to nodeapps.

Syntax and Options Use the `srvctl modify nodeapps` command with the following syntax:

```
srvctl modify nodeapps -n <node_name> [-o <oracle_home>] [-A <new_vip_address>]
```

Table B–38 *srvctl modify nodeapps Option*

Option	Description
-n <node_name>	Node name.
-o <oracle_home>	Oracle home for the cluster database.
-A <new_vip_address>	The node level VIP address (<name ip>/netmask[/if1[if2 ...]]).

Example An example of this is:

```
srvctl modify nodeapps -n mynode1 -A 100.200.300.40/255.255.255.0
```

srvctl relocate

Relocates the named service names from one named instance to another named instance. The `srvctl relocate` command works on only one source instance and one target instance at a time, relocating a service from a single source instance to a single target instance. The target instance must be on the preferred or available list for the service. The relocated service is temporary until you modify the configuration. The [srvctl modify](#) command described on page B-23 permanently changes the service configuration.

Table B–39 *srvctl relocate Summary*

Command	Description
srvctl relocate service on page B-24	Relocates the named service names from one named instance to another named instance

srvctl relocate service

Temporarily relocates a service member to run on another instance.

Syntax and Options Use the `srvctl relocate service` command with the following syntax:

```
srvctl relocate service -d <db_unique_name> -s <service_name> -i <old_inst_name>
-t <new_inst_name> [-f]
```

Table B–40 *srvctl relocate service Options*

Option	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	Service name.
-i <old_inst_name>	Old instance name.
-t <new_inst_name>	New instance name.
-f	Disconnect all sessions during stop or relocate service operations.

Example To temporarily relocate a named service member from `crm1` to `crm3`:

```
srvctl relocate service -d crm -s crm -i crm1 -t crm3
```

srvctl status

Displays the current state of a named database, instances, services, or node applications.

Table B–41 *srvctl status Summary*

Command	Description
srvctl status database on page B-24	Obtains the status of a database.
srvctl status instance on page B-25	Obtains the status of a instance.
srvctl status service on page B-25	Obtains the status of services.
srvctl status nodeapps on page B-26	Obtains the status of node applications.
srvctl status asm on page B-26	Obtains the status of ASM instances.

srvctl status database

Obtains the status of instances and their services.

Syntax and Options Use the `srvctl status database` command with the following syntax:

```
srvctl status database -d <db_unique_name> [-f] [-v]
```

Table B–42 *srvctl status database Options*

Option	Description
-d <db_unique_name>	Unique name for the database
-f	Include disabled applications
-v	Verbose output

Example An example of this command is:

```
srvctl status database -d crm -v
```

srvctl status instance

Obtains the status of instances.

Syntax and Options Use the `srvctl status instance` command with the following syntax:

```
srvctl status instance -d <db_unique_name> -i <inst_name_list> [-f] [-v]
```

Table B–43 *srvctl status instance Options*

Option	Description
-d <db_unique_name>	Unique name for the database
-i <inst_name_list>	Comma-separated list of instance names
-f	Include disabled applications
-v	Verbose output

Example An example of this command is:

```
srvctl status instance -d crm -i crm1,crm2 -v
```

srvctl status service

Obtains the status of a service.

Syntax and Options Use the `srvctl status service` command with the following syntax:

```
srvctl status service -d <db_unique_name> -s <service_name_list> [-f] [-v]
```

Table B–44 *srvctl status service Options*

Option	Description
-d <db_unique_name>	Unique name for the database
-s <service_name_list>	Comma-separated list of service names
-f	Include disabled applications
-v	Verbose output

Example The following example obtains the status of a named service globally across the database:

```
srvctl status service -d crm -s crm -v
```

srvctl status nodeapps

Obtains the status of node applications on a particular node.

Syntax and Options Use the `srvctl status nodeapps` command with the following syntax:

```
srvctl status nodeapps -n <node_name>
```

The only option available for this command is `-n` to specify the node name.

Example An example of this command to obtain the status of all nodes supporting database applications is:

```
srvctl status nodeapps -n mynode1
```

srvctl status asm

Obtains the status of an ASM instance.

Syntax and Options Use the `srvctl status asm` command with the following syntax:

```
srvctl status asm -n <node_name>
```

The only option available for this command is `-n` to specify the node name.

Example An example of this command is:

```
srvctl status asm -n crmnode1
```

srvctl getenv

Gets and displays values for the environment from the configuration file. Use SRVCTL with the `set`, `get`, and `unset` environment configuration verbs to administer the environment configurations for databases, instances, services, and node applications.

Table B-45 *srvctl getenv Summary*

Command	Description
srvctl getenv database on page B-26	Gets the cluster database environment values.
srvctl getenv instance on page B-27	Gets the instance environment values.
srvctl getenv service on page B-27	Gets the service environment values.
srvctl getenv nodeapps on page B-28	Gets the node application environment values.

srvctl getenv database

Displays the cluster database environment values.

Syntax and Options Use the `srvctl getenv database` command with the following syntax:

```
srvctl getenv database -d <db_unique_name> [-t <name_list>]
```

Table B–46 *srvctl getenv database Options*

Options	Description
-d <db_unique_name>	Unique name for the database
-t <name_list>	Names of environment variables

Example The following example gets the environment configuration for a cluster database:

```
srvctl getenv database -d crm
```

srvctl getenv instance

Gets the values for an instance environment configuration.

Syntax and Options Use the `srvctl getenv instance` command with the following syntax:

```
srvctl getenv instance -d <db_unique_name> -i <inst_name> [-t <name_list>]
```

Table B–47 *srvctl getenv database Options*

Options	Description
-d <db_unique_name>	Unique name for the database
-i <inst_name>	Instance name
-t <name_list>	Names of environment variables

Example The following example sets the environment configuration for an instance:

```
srvctl getenv instance -d -crm -i instance1
```

srvctl getenv service

Gets the values for a service environment configuration.

Syntax and Options Use the `srvctl getenv service` command with the following syntax:

```
srvctl getenv service -d <db_unique_name> -s <service_name> [-t <name_list>]
```

Table B–48 *srvctl getenv service Options*

Options	Description
-d <db_unique_name>	Unique name for the database
-s <service_name>	Service name
-t <name_list>	Names of environment variables

Example The following example lists all environment variables for a service:

```
srvctl getenv service -d crm -s crm
```

srvctl getenv nodeapps

Gets the environment variables for the node application configurations.

Syntax and Options Use the `srvctl getenv nodeapps` command with the following syntax:

```
srvctl getenv nodeapps -n <node_name> [-t <name_list>]
```

Table B–49 *srvctl getenv nodeapps Options*

Options	Description
-n <node_name>	Node name
-t <name_list>	Names of environment variables

Example The following example lists all environment variables for the node applications:

```
srvctl getenv nodeapps -n crmnode1
```

srvctl setenv and unsetenv

The `setenv` command sets values for the environment in the configuration file. The `unsetenv` command unsets values for the environment in the configuration file.

Table B–50 *srvctl setenv and unsetenv Summary*

Command	Description
srvctl setenv database on page B-28	Administers cluster database environment configurations
srvctl setenv instance on page B-29	Administers instance environment configurations
srvctl setenv service on page B-29	Administers service environment configurations
srvctl setenv nodeapps on page B-30	Administers node application environment configurations
srvctl unsetenv database on page B-30	Unsets the cluster database environment configuration
srvctl unsetenv instance on page B-30	Unsets instance environment configurations
srvctl unsetenv service on page B-31	Unsets service environment configurations
srvctl unsetenv nodeapps on page B-31	Unsets node application environment configurations

srvctl setenv database

Administers cluster database environment configurations.

Syntax and Options Use the `srvctl setenv database` command with the following syntax:

```
srvctl setenv database -d <db_unique_name> {-t <name>=<val>[,<name>=<val>,...] |
-T <name>=<val>}
```


Table B-51 *srvctl setenv database Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-t <name>=<val>,...	Names and values of environment variables.
-T <name>=<val>	Enables single environment variable to be set to a value that contains commas or other special characters.

Example The following example sets the language environment configuration for a cluster database:

```
srvctl setenv database -d crm -t LANG=en
```

srvctl setenv instance

Administers instance environment configurations.

Syntax and Options Use the `srvctl setenv instance` with the following syntax:

```
srvctl setenv instance -d <db_unique_name> [-i <inst_name>] {-t <name>=<val>[,<name>=<val>,...] | -T <name>=<val>}
```

Table B-52 *srvctl setenv instance Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-i <inst_name>	Instance name.
-t <name>=<val>,...	Names and values of environment variables.
-T <name>=<val>	Enables single environment variable to be set to a value that contains commas or other special characters.

Example The following example sets the environment configuration for an instance:

```
srvctl setenv instance -d -crm -i instance1 -t LANG=EN
```

srvctl setenv service

Administers service environment configurations.

Syntax and Options Use the `srvctl setenv service` command with the following syntax:

```
srvctl setenv service -d <db_unique_name> [-s <service_name>] {-t <name>=<val>[,<name>=<val>,...] | -T <name>=<val>}
```

Table B-53 *srvctl setenv service Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	Service name.
-t <name>=<val>,...	Names and values of environment variables.

Table B–53 (Cont.) *srvctl setenv service Options*

Options	Description
-T <name>=<val>	Enables single environment variable to be set to a value that contains commas or other special characters.

Example To set all environment variables for a service:

```
srvctl setenv service -d crm -s crm -t CLASSPATH=/usr/local/jdk/jre/rt.jar
```

srvctl setenv nodeapps

Sets the environment variables for the node application configurations.

Syntax and Options Use the `srvctl setenv nodeapps` command as follows:

```
srvctl setenv nodeapps -n <node_name> {-t <name>=<val>[,<name>=<val>,...] | -T <name>=<val>}
```

Table B–54 *srvctl setenv nodeapps Options*

Options	Description
-n <node_name>	Node name.
-t <name>=<val>,...	Names and values of environment variables.
-T <name>=<val>	Enables single environment variable to be set to a value that contains commas or other special characters.

Example To set an environment variable for a node application:

```
srvctl setenv nodeapps -n crmnode1 -t CLASSPATH=/usr/local/jdk/jre/rt.jar
```

srvctl unsetenv database

Unsets the cluster database environment configurations.

Syntax and Options Use the `srvctl unsetenv database` command as follows:

```
srvctl unsetenv database -d <db_unique_name> -t <name_list>
```

Table B–55 *srvctl unsetenv database Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-t <name_list>	Names of environment variables.

Example The following example unsets the environment configuration for a cluster database environment variable:

```
srvctl unsetenv database -d crm -t CLASSPATH
```

srvctl unsetenv instance

Unsets instance environment configurations.

Syntax and Options Use the `srvctl unsetenv instance` command as follows:

```
srvctl unsetenv instance -d <db_unique_name> [-i <inst_name>] -t <name_list>
```

Table B-56 *srvctl unsetenv instance Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-i <instance_name>	Instance name.
-t <name_list>	Names of environment variables.

Example The following example unsets the environment configuration for an instance:

```
srvctl unsetenv instance -d -crm -i instance1 -t CLASSPATH
```

srvctl unsetenv service

Unsets service environment configurations.

Syntax and Options Use the `srvctl unsetenv service` command as follows:

```
srvctl unsetenv service -d <db_unique_name> [-s <service_name>] -t <name_list>
```

Table B-57 *srvctl unsetenv service Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	Service name.
-t <name_list>	Names of environment variables.

Example To unset an environment variables for a service:

```
srvctl unsetenv service -d crm -s crm -t CLASSPATH
```

srvctl unsetenv nodeapps

Unsets the environment configuration for the node application configurations.

Syntax and Options Use the `srvctl unsetenv nodeapps` command as follows:

```
srvctl unsetenv nodeapps -n <node_name> -t <name_list>
```

Table B-58 *srvctl unsetenv nodeapps Options*

Options	Description
-n <node_name>	Node name.
-t <name>=<val>	Names and values of environment variables.

Example The following example unsets the environment configuration for a node's node applications:

```
srvctl unsetenv nodeapps -n crmmodel -t <name_list>
```

srvctl remove

Removes the configuration, the CRS applications for the node (including the virtual IP address, the Oracle Enterprise Manager agent, the GSD, and the listeners), the database, named instances, or the named services from the cluster database. Environment settings for the object are also removed. If you do not use the force flag (-f), then Oracle prompts you to confirm whether to proceed. If you use the force (-f) option, then the remove operation proceeds without prompting. To use the `remove` verb, you must first stop the node applications, database, instance, or service for which you are specifying `srvctl remove`. Oracle recommends that you perform a disable operation before using this command, but this is not required. You must stop the target object before running the `srvctl remove` command. The [srvctl stop](#) command is described on page B-17.

Table B-59 *srvctl remove Summary*

Command	Description
srvctl remove database on page B-32	Removes a database and configuration.
srvctl remove instance on page B-32	Removes one or more instances and configurations.
srvctl remove service on page B-33	Removes services.
srvctl remove nodeapps on page B-33	Removes node applications.
srvctl remove asm on page B-34	Removes ASM instances

srvctl remove database

Removes a database configuration.

Syntax and Options Use the `srvctl remove database` command with the following syntax:

```
srvctl remove database -d <db_unique_name> [-f]
```

Table B-60 *srvctl remove database Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-f	Force remove.

Example An example of this command is:

```
srvctl remove database -d crm
```

srvctl remove instance

Removes the configurations for an instance.

Syntax and Options Use the `srvctl remove instance` command with the following syntax:

```
srvctl remove instance -d <db_unique_name> -i <inst_name> [-f]
```

Table B–61 *srvctl remove instance Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-i <inst_name>	Instance name.
-f	Force remove.

Example An example of this command is:

```
srvctl remove instance -d crm -i crm01
```

srvctl remove service

Removes the configuration for a service.

Syntax and Options Use the `srvctl remove service` command as follows:

```
srvctl remove service -d <db_unique_name> -s <service_name> [-i <inst_name>] [-f]
```

Table B–62 *srvctl remove service Options*

Options	Description
-d <db_unique_name>	Unique name for the database.
-s <service_name>	Service name.
-i <inst_name>	Instance name.
-f	Force remove.

Examples An example of this command is:

```
srvctl remove service -d crm -s sales
```

The following example removes the services from specific instances:

```
srvctl remove service -d crm -s sales -i crm01,crm02
```

srvctl remove nodeapps

Removes the node application configuration from the specified node. You must have full administrative privileges to run this command. On UNIX-based systems, you must be logged in as `root` and on Windows-based systems, you must be logged in as a user with Administrator privileges.

Syntax Use the `srvctl remove nodeapps` command as follows:

```
srvctl remove nodeapps -n <node_name_list> [-f]
```

Table B–63 *srvctl remove nodeapps Options*

Options	Description
-n <node_name_list>	Node name or a comma-separated list of node names.
-f	Force remove.

Example An example of this command is:

```
srvctl remove nodeapps -n mynode1,mynode2,mynode3
```

srvctl remove asm

Removes an ASM instance.

Syntax and Options Use the `srvctl remove asm` command with the following syntax:

```
srvctl remove asm -n <node_name> [-i <asn_inst_name>]
```

Table B-64 *srvctl remove asm Option*

Option	Description
-n <node_name>	Node name
-i <asm_inst_name>	ASM instance name.

Example An example of this command is:

```
srvctl remove asm -n crmnode1 -i asm1
```

Oracle Real Application Clusters Tools Messages

This appendix describes the Oracle Real Application Clusters (RAC) management tools messages. The messages in this appendix appear alphabetically by message prefix in the following groups:

- PRKA—Cluster Node Applications Messages
- PRKC—Cluster Command Messages
- PRKD — Global Services Daemon Messages
- PRKE — Global Services Daemon Controller Utility Messages
- PRKH—Server Manager (SRVM) Messages
- PRKN— Server Manager (SRVM) System Library Messages
- PRKO—Server Control (SRVCTL) Utility Messages
- PRKP—Cluster Database Management Messages
- PRKR—Cluster Registry Messages
- PRKS—Automatic Storage Management Messages
- PRKU—Command Line Parser Utility Messages
- PRKV — Virtual IP Configuration Assistant Messages

See Also: *Oracle Database Platform Guide for Windows* for Windows-based messages and for all other messages refer search online at

<http://tahiti.oracle.com>

Overview of Real Application Clusters-Specific Messages

Within Oracle Database 10g, RAC messages use the same ORA prefix and reside in the same e*.msg files as non-RAC or single-instance messages. Additionally, some RAC messages are issued by the single-instance Oracle database. These messages appear online by way of a Tahiti error message search as described in the Oracle database documentation in the Preface of this book under the heading "[Related Documents](#)" on page -xiii.

Prefixes and Message Codes for RAC-Specific Messages

Message prefixes indicate where to find information about the messages in this chapter. In addition, the prefixes indicate which RAC component issued the messages.

Types of Real Application Clusters Messages and Related Files

The RAC high availability and cluster database management tools use the following syntax for internal messages:

```
PRKx-0000: Cause description "variable" text {variable} text {variable}
```

A message might appear as follows:

```
PRKC-1022 "Could not get "node name" for node {0} in {1}"
```

The variables in the previous example represent numbers, names, and character strings for objects in the physical or logical cluster and the cluster database. you can ignore empty variables.

PRKA—Cluster Node Applications Messages

PRKA-2001: GSD already exists

Cause: An attempt was made to create the Global Services Daemon (GSD) application while there is a running GSD application.

Action: Stop the running GSD, delete it and create the GSD application again using 'srvctl add nodeapps'

PRKA-2002: Listener already exists

Cause: An attempt was made to create a Listener application while there is a running Listener application.

Action: Stop the running Listener, delete it and create the Listener application again using 'srvctl add nodeapps'

PRKA-2010: VIP already exists

Cause: An attempt was made to create a Virtual IP while there is a running Virtual IP application.

Action: Stop the running Virtual IP application, delete it and create the Virtual IP application using 'srvctl add nodeapps'

PRKC—Cluster Command Messages

PRKC-1000 "No active nodes detected in the cluster"

Cause: The active node list of the cluster could not be retrieved. This can occur if the clusterware was not functioning properly in one or more nodes in the cluster, or because there was no clusterware installed on the machine.

Action: Check the state of the cluster by running 'olsnodes' binary from your <crs home>/bin and see if it prints the node list correctly.

PRKC-1001 "Error submitting commands in the buffer"

Cause: An internal error has occurred. This error is not exposed to the clients.

Action: Contact your customer support representative.

PRKC-1002 "All the submitted commands did not execute successfully"

Cause: This could happen either because some node in the cluster failed in the middle of a manageability operation or because the communication channel between nodes failed.

Action: This could happen either because some node in the cluster failed in the middle of a manageability operation or because the communication channel between nodes failed.

PRKC-1004 "Problem in copying file to nodes"

Cause: Attempted file copy operation(s) from the local node to one or more nodes in the cluster and one or multiple of those copy file operations failed. The possible causes are that one or more nodes failed during the operation, the destination directory did not have write permission for the user on one or more of the nodes, the source file did not exist, or the 'rcp/rsh/scp/ssh' command to one or more nodes failed on UNIX platforms.

Action: Check that the source file exists. Check that all nodes in the cluster are up. Check that destination directory had write permission for the user. On UNIX based platforms check that user can do 'rcp/rsh/scp/ssh' to all the nodes.

PRKC-1005 "Problem in removing file from nodes"

Cause: Attempted file remove operation(s) from the local node to one or more nodes in the cluster and one or multiple of those file remove operations failed. The possible causes are that one or more nodes failed during the operation, the destination directory did not have write permission for the user on one or more of the nodes, the source file did not exist, or the 'rcp/rsh/scp/ssh' command to one or more nodes failed on UNIX platforms.

Action: See earlier error message for details.

PRKC-1006 "Problem in moving file to nodes"

Cause: Attempted file move operation(s) from the local node to one or more nodes in the cluster and one or multiple of those move file operations failed. The possible causes are that one or more nodes failed during the operation, the destination directory did not have write permission for the user on one or more of the nodes, the source file did not exist, or the 'rcp/rsh/scp/ssh' command to one or more nodes failed on UNIX platforms.

Action: See earlier error message for details.

PRKC-1007 "Problem in creating directories on the nodes"

Cause: Attempted directory create operation(s) from the local node to one or more nodes in the cluster and one or multiple of those directory create operations failed. The possible causes are that one or more nodes failed during the operation, the destination directory did not have write permission for the user on one or more of the nodes, the source file did not exist, or the 'rcp/rsh/scp/ssh' command to one or more nodes failed on UNIX platforms.

Action: See earlier error message for details.

PRKC-1009 "Failed to start the service on all nodes"

Cause: Attempted to start a service on all the nodes in the cluster and one or multiple of those services did not start successfully.

Action: Check if all the nodes in cluster are up.

PRKC-1010 "Failed to stop the service on all nodes"

Cause: Attempted to stop a service on all the nodes in the cluster and one or multiple of those services did not start successfully.

Action: Check if all the nodes in cluster are up.

PRKC-1011 "Failed to delete the service from all nodes"

Cause: Attempted to delete a service on all the nodes in the cluster and one or multiple of those services did not start successfully.

Action: Check if all the nodes in cluster are up.

PRKC-1012 "Failed to create the service on all node"

Cause: Attempted to delete a service on all the nodes in the cluster and one or multiple of those services did not start successfully.

Action: Check if all the nodes in cluster are up.

PRKC-1016 "Problem in retrieving value of the enviornment variable"

Cause: Attempted to retrieve value of an environment variable which was not defined.

Action: Define the environment variable.

PRKC-1017 "Problem restarting GSD"

Cause: GSD (Global Services Daemon) could not be started on the local node. This can occur if the clusterware was not functioning properly or there is already a running GSD.

Action: Check the state of your clusterware by running 'lsnodes' from <crs home>/bin. It should list the nodes in the cluster. Try 'gsdctl stop' and then 'gsdctl start' again.

PRKC-1018 "Error getting coordinator node"

Cause: This can occur if the clusterware was not functioning properly.

Action: Use 'gsdctl stop' on each node to stop the GSDs. Use 'gsdctl start' on each node in the cluster.

PRKC-1019 "Error creating handle to daemon on the node {0}"

Cause: Global Services Daemon was not running on the node.

Action: Use 'gsdctl stat' to check the status of the daemon. Use 'gsdctl start' to start it.

PRKC-1020 "Exception while executing the operation on the remote node {0}"

Cause: Attempted to execute an operation on the remote node when the Global Services Daemon was not up on the remote node {0}.

Action: Start the Global Services Daemon on the remote node using 'gsdctl start'.

PRKC-1021 "Problem in the clusterware"

Cause: The clusterware cannot be contacted. This can occur because the clusterware was not functioning properly.

Action: Check the state of the clusterware.

PRKC-1022 "Could not get "node name" for node {0} in {1}"

Cause: The clusterware cannot be contacted. This can occur because the clusterware was not functioning properly.

Action: Check the state of the clusterware using 'olsnodes -n'.

PRKC-1023 "Invalid IP address format: {0}"

Cause: The given IP address is not valid.

Action: Check the format of the IP address.

PRKC-1024 "Invalid netmask: {0}"

Cause: Netmask passed was invalid.

Action: Check that it was a valid netmask.

PRKC-1025 "Failed to create a file in directory {0}"

Cause: Attempted file create operation(s) from the local node to one or more nodes in the cluster and one or multiple of those copy file operations failed. The possible causes are that one or more nodes failed during the operation, the destination directory did not have write permission for the user on one or more of the nodes, the source file did not exist, or the 'rcp/rsh/scp/ssh' command to one or more nodes failed on UNIX platforms.

Action: Check that all nodes in the cluster are up. Check that destination directory had write permission for the user. On UNIX based platforms check that user can do 'rcp/rsh/scp/ssh' to all the nodes.

PRKC-1026 "Failed doing I/O to file {0}"

Cause: Attempted to write to or read from a file.

Action: Check the file exists. Check that the file has read/write permission for the user.

PRKC-1027 "Error checking existence of file {0} on {1}"

Cause: Attempted to check the existence of a file on a given node.

Action: Check that the user has execute permission on the given directory. On UNIX based platforms check that user can do 'rcp/rsh/scp/ssh' to the given node.

PRKC-1028 "Error checking write permission for directory {0} on {1}"

Cause: Attempted to check the write permission for the given directory on the given node.

Action: On UNIX based platforms check that user can do 'rcp/rsh/scp/ssh' to the given node.

PRKC-1029 "Failed to get modification time for file {0} on {1}"

Cause: Cannot do rsh/ssh to the given node.

Action: On UNIX based platforms check that user can do 'rcp/rsh/ssh/scp' to the given node.

PRKC-1030 "Error checking accessibility for node {0}"

Cause: Cannot do rsh/ssh to the given node.

Action: On UNIX based platforms check that user can do 'rcp/rsh/ssh/scp' to the given node.

PRKC-1031 "Error checking free space for {0} on {1}"

Cause: Cannot get free amount of available free space on a given node.

Action: On UNIX based platforms check that user can do 'rcp/rsh/ssh/scp' to the given node.

PRKC-1032 "Directory {0} does not exist"

Cause: Attempted an operation on a directory which does not exist.

Action: Make sure that the directory exists.

PRKC-1033 "Executable {0} does not exist"

Cause: Attempted an invoke the given executable.

Action: Make sure that the executable exists.

PRKC-1034 "No local node name found for host {0}"

Cause: Could not find the node name for a given host.

Action: Contact your local customer service representative.

PRKC-1035 "Node names for this cluster could not be retrieved"

Cause: Could not retrieve the node names for the Oracle clusterware. This could be caused if the Oracle clusterware was never installed on the system or if your Oracle Cluster Registry got corrupted.

Action: Install Oracle clusterware. Contact your local customer service representative

PRKC-1036 "CRS_HOME name passed to the method was null"

Cause: ORA_CRS_HOME retrieved from Oracle Cluster Registry was null.

Action: Contact your local customer service representative.

PRKC-1037 "Error removing files listed in {0} from node {1}"

Cause: Attempted to remove the given file from the given node.

Action: Check that all nodes in the cluster are up. Check that destination directory had write permission for the user. On UNIX based platforms check that user can do 'rcp/rsh/scp/ssh' to all the nodes.

PRKC-1038 "Error copying files listed in {0} to node {1}"

Cause: Attempted file create operation(s) from the local node to one or more nodes in the cluster and one or multiple of those copy file operations failed. The possible causes are, one or more nodes failed during the operation, the destination directory did not have write permission for the user on one or more of the nodes, the source file did not exist, or the 'rcp/rsh/scp/ssh' command to one or more nodes failed on UNIX platforms.

Action: Check that all nodes in the cluster are up. Check that destination directory had write permission for the user. On UNIX based platforms check that user can do 'rcp/rsh/scp/ssh' to all the nodes.

PRKC-1039 "Error creating directories listed in {0} on node {1}"

Cause: Attempted directory create operation(s) from the local node to one or more nodes in the cluster and one or multiple of those copy file operations failed. The possible causes are that one or more nodes failed during the operation, the destination directory did not have write permission for the user on one or more of the nodes, the source file did not exist, or the 'rcp/rsh/scp/ssh' command to one or more nodes failed on UNIX platforms.

Action: Check that all nodes in the cluster are up. Check that destination directory had write permission for the user. On UNIX based platforms check that user can do 'rcp/rsh/scp/ssh' to all the nodes.

PRKC-1040 "Remote Shell is not known yet."

Cause: Cannot determine the location of your remote shell command (rsh/ssh).

Action: Make sure you have rsh/ssh program installed on your system.

PRKC-1041 "Remote Copy command is not known yet."

Cause: Cannot determine the location of your remote copy command (rcp/scp).

Action: Make sure you have rcp/scp program installed on your system.

PRKC-1042 "The Remote Shell {0} requested by client is not recognized "

Cause: Cannot determine the location of your remote shell command (rcp/scp).

Action: Make sure you have rsh/ssh program installed on your system.

PRKC-1043 "The Remote Copy command {0} requested by client is not recognized"

Cause: Cannot determine the location of your remote copy command (rcp/scp).

Action: Make sure you have rcp/scp program installed on your system.

PRKC-1044 "Failed to check remote command execution setup for node {0} using shells {1} and {2} "

Cause: Internal error.

Action: Contact your local customer service representative. PRKC-1045 "Node name passed to the method was null."

PRKC-1046 "The file list passed to the method was null"

Cause: Internal error.

Action: Contact your local customer service representative.

PRKC-1047 "The directory list passed to the method was null"

Cause: Internal error.

Action: Contact your local customer service representative.

PRKC-1048 "{0} is not supported on Windows platform"

Cause: Attempted to use a facility on Windows which is not supported.

Action: Contact your local customer service representative.

PRKC-1049 "CRS is not supported in version {0}. It is supported from version {1} onwards"

Cause: Attempted to use cluster ready services facilities from a program which is at a prior version. This is an internal error.

Action: Contact your local customer service representative.

PRKC-1050 "EVM is not supported in version {0}. It is supported from version {1} onwards"

Cause: Attempted to use cluster ready services facilities from a program which is at a prior version. This is an internal error.

Action: Contact your local customer service representative.

PRKC-1051 "Invalid data to set for registry key {0}."

Cause: Attempted to set data unrecognized by Windows operating system to a Windows registry key.

Action: Contact your local customer service representative.

PRKC-1052 "Invalid data type for registry key {0}."

Cause: Attempted to set an invalid data type to a Windows registry key.

Action: Contact your local customer service representative.

PRKC-1053 "Error returned from node {0} is \"{1}\""

Cause: An error was returned.

Action: Contact your local customer service representative.

PRKC-1054 "Node {0} is not accessible"

Cause: One of the nodes on the cluster is not accessible.

Action: Check the network connectivity.

PRKC-1055 "Directory name passed was null"

Cause: An internal error occurred.

Action: Contact your customer support representative.

PRKC-1056 "Failed to get the hostname for node {0}"

Cause: Couldn't retrieve the host name from one of the nodes.

Action: Make sure hostnames are configured on all nodes.

PRKC-1057 "The computername and hostname do not match for node {0}. Make sure that they match."

Cause: The computer name and the hostname didn't match on one of the nodes.

Action: Make sure that they match.

PRKC-1058 "The hostname registry key {0} has an empty value on node {1}"

Cause: An internal error occurred.

Action: Contact your customer support representative.

PRKC-1059 "The computername registry key {0} has an empty value on node {1}"

Cause: An internal error occurred.

Action: Contact your customer support representative.

PRKC-1060 "Cluster misconfigured on node \"{0}\", \"{1}\" property in {2} does not match with configured data in Oracle Cluster Registry"

Cause: Cluster is misconfigured on the given node.

Action: Contact your customer support representative.

PRKD — Global Services Daemon Messages

PRKD-3000 "Failed to initialize and register with clusterware"

Cause: This can occur if cluster synchronization services was not functioning properly.

Action: Check the state of your clusterware by running olsnodes from ORA_CRS_HOME/bin. It should list the nodes in the cluster.

PRKD-3001 "Ready to receive client requests"

Cause: Global Services Daemon is ready.

Action: No action needed at this time.

PRKE — Global Services Daemon Controller Utility Messages

PRKE-1008 "Failed to get list of active nodes from clusterware"

Cause: Not operating in cluster mode, failure to load srvm libraries, failure to initialize clusterware context.

Action: Check for the health of clusterware using "lsnodes" utility to get the list of active nodes in the cluster. Also, check for the presence of srvm libraries in the load library path. If these two checks get cleared, please contact your customer support representative.

PRKE-1009 "Failed to start GSD on local node"

Cause: GSD is already running on the local node, or GSD could not be started because of some other reason.

Action: Run "gsdctl stat" to check if a daemon is already running on the local node. If a daemon is not running on local node, please contact your customer support representative to check for the other reasons for failure.

PRKE-1010 "Failed to stop GSD on local node"

Cause: GSD is not running on the local node, or GSD could not shutdown gracefully.

Action: Run "gsdctl stat" to check if a GSD is running on the local node. If a GSD is running, please contact your customer support representative to check for other reasons for failure.

PRKE-1011 "Failed to get status of GSD on local node"

Cause: Failure to get the list of active nodes from the cluster, or failure to query the list of live daemons from the clusterware.

Action: Run "lsnodes" to check the health of the clusterware. If the clusterware is working fine, check for the presence of srvm libraries in the load library path. If the libraries are present, please contact your customer support representative for further assistance.

PRKH—Server Manager (SRVM) Messages

PRKH-1000: "Unable to load the SRVM HAS shared library"

Cause: The system cannot find, or load the srvm has shared library.

Action: Check your system's library load path. Verify that the library exists, and that the libraries it depends on are readable and are in the load path.

PRKH-1001: "HASContext Internal Error"

Cause: An unexpected internal error has occurred while attempting to communicate with CRS.

Action: Contact Oracle support

PRKH-1002: "Internal HASContext Error: JNI Native Call Failure"

Cause: An unexpected internal error has occurred while executing native code from java.

Action: Contact Oracle support

PRKH-1003: "Failed to allocate memory in native layer: "

Cause: The SRVM framework was unable to allocate memory.

Action: Your system is running low on memory. Check the memory and swap space, resolve the problem and retry.

PRKH-1004: "Failed to execute remote join cluster alias {0} for nodes: "

Cause: Unable to execute a join cluster alias operation on another node.

Action: Check the failed nodes. Verify that the system can execute remote commands on those nodes. Check the cluster alias and it's related functionality.

PRKH-1005: "Failed to execute remote leave cluster alias {0} for nodes: "

Cause: Unable to execute a leave cluster alias operation on another node.

Action: Check the failed nodes. Verify that the system can execute remote commands on those nodes. Check the cluster alias and it's related functionality.

PRKH-1006: "The following nodes are either not active or not configured: "

Cause: The system was unable to find a node requested.

Action: Please check the node list supplied and verify that the node is available and configured.

RKH-1007: "Exception Caused by: "

Cause: The current exception was caused by another earlier exception.

Action: Examine all of the nested exceptions to determine the root cause of the error.

PRKH-1008: "Internal HASContext Error: Argument {0} must be set."

Cause: Internal software error.

Action: Contact Oracle support.

PRKH-1009: "CRS HOME must be defined in the environment or in the Oracle Cluster Registry"

Cause: The system is unable to determine the ORA_CRS_HOME for this CRS installation.

Action: Check that the Oracle Cluster Registry is properly configured and available to this node.

PRKH-1010: "Unable to communicate with CRS services."

Cause: The system is unable to communicate with the CRS services.

Action: Check that all the CRS daemons are running and are properly configured. Verify that the current program is attempting to communicate with the correct CRS daemons.

PRKH-1011: "Process does not have sufficient privileges to perform the requested operation. {0}"

Cause: The user running the current operation is not permitted to execute a given operation.

Action: Check other messages in the error stack to determine which operation failed. Verify that you are running the operation as the correct user.

PRKH-1012: "Unable to find or resolve user {0}."

Cause: The system was unable to find the username supplied.

Action: Check the username and try again.

PRKH-1013: "The oracle home {0} does not exist."

Cause: The system was unable to find the oracle home supplied.

Action: Check the full path of the oracle home and try again.

PRKN— Server Manager (SRVM) System Library Messages

PRKN-1008: Unable to load the shared library "{0}" or a dependent library, from {1}={2} [{3}]

Cause: The SRVM framework was unable to load a shared library to perform native operations.

Action: Make sure that the library in question is installed and in a location where it can be loaded. Check the environment setting that determines shared library loading on your platform.

PRKN-1009: Native System Internal Error

Cause: An unexpected internal error occurred while attempting to execute a native operation from Java.

Action: Contact Oracle Support.

PRKN-1010: This system is not properly configured as a cluster

Cause: SRVM was unable to find the system libraries which are required in order to use the system as a cluster.

Action: Verify that the CRS installation was done properly, try to execute 'srvctl status' command to get more diagnostic information.

PRKN-1011: Failed to retrieve value for "{0}" under registry key "{1}" on node "{2}", {3}

Cause: There was a problem accessing the Windows registry key.

Action: Open Windows registry editor on the node, check if the registry key exists and it contains the subkey.

PRKO—Server Control (SRVCTL) Utility Messages

PRKO-2001: "Invalid command line syntax"

Cause: An invalid SRVCTL command line was entered.

Action: Use -h SRVCTL command line option to find out the correct command line syntax and re-enter the command.

PRKO-2002: "Invalid command line option: "

Cause: An invalid SRVCTL command line option was entered.

Action: Use -h SRVCTL command line option to find out the correct command line syntax and re-enter the command.

PRKO-2003: "Invalid command line option value: "

Cause: An invalid SRVCTL command line option value was entered.

Action: Use -h SRVCTL command line option to find out the correct command line syntax and re-enter the command.

PRKO-2004: "Repetition of command line option: "

Cause: Duplicate SRVCTL command line option was entered.

Action: Eliminate the duplicate and re-enter the command.

PRKO-2005: "Application error: Failure in getting Cluster Database Configuration for: "

Cause: An error occurred while getting Cluster Database configuration for the named database.

Action: Run the 'srvctl config' command to make sure that the database has been configured. Run the 'srvctl add' command to add the configuration.

PRKO-2006: "Invalid node name: "

Cause: An invalid node name was entered.

Action: Use the correct node name. A valid node name should match the output from 'olsnodes' and must not contain domain name.

PRKO-2007: "Invalid instance name: "

Cause: An invalid instance name was entered.

Action: Use the correct instance name for the database. Run 'srvctl config database -d <db_name>' command to find out all instances of a database in the Cluster Database Configuration.

PRKO-2008: "Invalid connect string: "

Cause: An invalid connect string was entered.

Action: Use the correct connect string syntax: <user>/<password>[as <role>]

PRKO-2009: "Invalid name/value string: "

Cause: An invalid environment name/value pair was entered during SRVCTL setenv command.

Action: Make sure that the correct name/value string format is used: <name>=<value>. Try using quotes for values that contain spaces or special characters.

PRKO-2010: "Error in adding instance to node: "

Cause: An error occurred when adding instance to Cluster Database configuration.

Action: Use 'srvctl config database -d <db_name>' command to check if the database has been configured in Cluster Database Configuration Repository. If it is a permission issue, run as the user who created the database configuration.

PRKO-2011: "Error in removing instance: "

Cause: An error occurred when removing an instance from Cluster Database Configuration.

Action: Use 'srvctl config database -d <db_name>' command to check if the database and instance have been configured in Cluster Database Configuration Repository. If it is a permission issue, run as the user who created the database configuration.

PRKO-2012: "Error in moving instance to node: "

Cause: An error occurred when changing the instance and node mapping in Cluster Database Configuration.

Action: Use 'srvctl config database -d <db_name>' command to check if the database and instance have been configured in Cluster Database Configuration Repository. If it is a permission issue, run as the user who created the database configuration.

PRKO-2013: "Error in setting env: "

Cause: An error occurred when setting environment variables for a database or an instance.

Action: Use 'srvctl config database -d <db_name>' command to check if the database and/or the instance have been configured in Cluster Database Configuration Repository. If it is a permission issue, run as the user who created the database configuration.

PRKO-2014: "Error in unsetting env: "

Cause: An error occurred when unsetting environment variables for a database or an instance.

Action: Use 'srvctl getenv' command to check if the environment variable exists for the database or instance; make sure that the database and/or the instance have been configured in Cluster Database Configuration Repository. If it is a permission issue, run as the user who created the database configuration.

PRKO-2015: "Error in checking condition of instance on node: "

Cause: Could not get status information of an instance.

Action: Use 'srvctl config database -d <db_name>' command to check if the instance has been configured in the Oracle database. If it is a permission issue, run as the user who created the database configuration.

PRKO-2016: "Error in checking condition of listener on node: "

Cause: Could not get status information of a listener.

Action: Check if the listener has been configured in the listener configuration file through the attribute LISTENER_<nodename>.

PRKO-2017: Service {0} does not exist for database {1}.

Cause: The specified service name does not exist.

Action: Verify that the service name is correct. Run a 'srvctl add service' command for adding the service.

PRKO-2018: "Mutual exclusive command line options -c and -q cannot be used at the same time."

Cause: Mutually exclusive command line options specified on the command line.

Action: Use either -c or -q option but not both.

PRKO-2101: "Instance is disabled: "

Cause: Attempt to relocate a service on an instance that is disabled.

Action: Enable the instance by running the command 'srvctl enable instance'.

PRKO-2102: "Service {0} is not supported on instance {1}."

Cause: The specified instance is neither preferred instance nor available instance for the service.

Action: Try another instance that is configured for the service.

PRKO-2104: "Error in checking condition of service: "

Cause: Possible error with the crs_stat command or a permission error.

Action: Make sure CRS daemon is running; execute the crs_stat command to determine if the daemon responds.

PRKO-2105: "Error in checking condition of VIP on node: "

Cause: Cannot obtain status of VIP due to issues in CRS daemon.

Action: Make sure CRS daemon is running; execute the crs_stat command to determine if the daemon responds.

PRKO-2106: "Error in checking condition of GSD on node: "

Cause: Cannot obtain status of GSD due to issues in CRS daemon.

Action: Make sure CRS daemon is running; execute the crs_stat command to see if CRS daemon responds.

PRKO-2108: "Node applications are still running on node: "

Cause: Node-level applications are still running and cannot be removed.

Action: Stop node applications before removing.

PRKO-2109: "Invalid address string: "

Cause: Bad address string format.

Action: Execute the srvctl -h command to see the correct address string format; try again.

PRKO-2110: "Invalid IP range string: "

Cause: Bad IP range format.

Action: Execute the srvctl -h command to see the correct IP range format; try again.

PRKO-2111: "Failure in getting VIP range for the cluster"

Cause: Cannot get IP range due to OCR issue.

Action: Make sure the OCR is configured; make sure CRS daemon is running by running the crs_stat command.

PRKO-2112: "Some or all node applications are not removed successfully on node: "

Cause: Some node applications are not successfully removed due to issues in CRS daemon or the OCR.

Action: Make sure CRS daemon is running by running the crs_stat command; make sure OCR is configured.

PRKO-2113: "Instance {0} is already a preferred instance for service {1}."

Cause: Attempt to add an instance as a preferred instance which is already preferred.

Action: No action required.

PRKO-2114: "Instance {0} is already an available instance for service {1}."

Cause: Attempt to add an instance as a available instance which is already available.

Action: No action required.

PRKO-2115: "Cannot remove node-level applications on node {0}, because a listener application exists. Remove the listener application first, then re-run this command."

Cause: A listener application exists already.

Action: Remove the listener application using the Network Configuration Assistant or crs_unregister, then re-run this command.

PRKO-2116: "Error in checking condition of ONS daemon on node: "

Cause: Cannot obtain status of Oracle Notification Service daemon due to issues in CRS daemon.

Action: Make sure CRS daemon is running; execute the `crs_stat` command to determine if CRS daemon responds.

PRKO-2117: "This command should be executed as the system privileged user."

Cause: A privileged command is run by the non-privileged user.

Action: Obtain root access or become a member of the Administrators group on Windows.

PRKO-2118: "Error in checking condition of ASM instance {0} on node {1}."

Cause: Cannot obtain status of ASM instance due to issues in CRS daemon.

Action: Make sure CRS daemon is running; execute the `crs_stat` command to determine if CRS daemon responds.

PRKO-2119: "Error in checking enable/disable status of ASM instance {0} on node {1}."

Cause: Cannot obtain status of ASM instance due to issues in CRS daemon.

Action: Make sure CRS daemon is running; execute the `crs_stat` command to determine if CRS daemon responds.

PRKO-2120: "The internal database service {0} cannot be managed with srvctl."

Cause: Attempt to manage the internal database service with `srvctl`.

Action: Use another service name for this service.

PRKO-2318: Target instance {0} already supports service {1}.

Cause: Modify service failed because the target instance is already part of service configuration.

Action: Use an instance that does not support the service.

PRKO-2319: Instance {0} is not an available instance for service {1}.

Cause: Instance is not in service configuration.

Action: Pick an instance that supports the service.

PRKP—Cluster Database Management Messages

PRKP-1000 "Cannot retrieve configuration for cluster database {0}"

Cause: The cluster database configuration cannot be retrieved from the repository. This can occur either because the database was never registered, or because the repository itself has not been created.

Action: Check if the database has been configured by printing a list of all cluster databases using `'srvctl config'`. If the repository has not been created, use `'srvconfig -init'` to create it.

PRKP-1001 "Error starting instance {0} on node {1}"

Cause: The instance could not be started using the SQL*Plus startup command.

Action: Try starting the named instance manually using SQL*Plus to see why it failed.

PRKP-1002 "Error stopping instance {0} on node {1}"

Cause: The SQL*Plus shutdown command returned an error while stopping the instance.

Action: Try stopping the named instance manually using SQL*Plus to see why it failed.

PRKP-1003 "Startup operation partially failed"

Cause: Some components of the cluster database could not be started.

Action: See earlier error message for details

PRKP-1004 "Shutdown operation partially failed"

Cause: Some components of the cluster database reported errors while being stopped

Action: See earlier error messages for details

PRKP-1005 "Failed to start up cluster database {0}"

Cause: The cluster database could not be started.

Action: See earlier error messages for details.

PRKP-1006 "Failed to shut down cluster database {0}"

Cause: The cluster database reported errors while being shut down

Action: See earlier error messages for details

PRKP-1007 "Failed to start all the listeners associated with all the instances of cluster database {0}"

Cause:

Action: Contact your customer support representative.

PRKP-1008 "Failed to start listeners associated with instance {0} on node {1}"

Cause:

Action: Contact your customer support representative

PRKP-1009 "Failed to stop all the listeners associated with all the instances of cluster database {0}"

Cause: Either the listener name associated with an instance could not be determined, or "lsnrctl stop" failed for a listener.

Action: Verify that listener.ora contains a SID_LIST entry for each instance of the named database, and that the lsnrctl stop command succeeds for those listeners

PRKP-1010 "Failed to stop all the listeners associated with instance {0} on node{1}"

Cause: Either the listener name associated with an instance could not be determined, or "lsnrctl stop" failed for a listener.

Action: Verify that listener.ora contains a SID_LIST entry for each instance of the named database, and that the lsnrctl stop command succeeds for those listeners

PRKP-1011 "Failed to get status of the listeners associated with instance {0} on node{1}"

Cause: Either the listener name associated with the instance could not be determined, or "lsnrctl status" failed for that listener.

Action: Verify that listener.ora contains a SID_LIST entry for the named instance, and that the lsnrctl status command succeeds

PRKP-1012 "Invalid environment variable {0} setting for cluster database {1}"

Cause: The argument to the -t option is not of the form <name>=<value> or it contains special characters.

Action: Ensure that the -t option has an argument of the form <name>=<value>. Enclose the argument to the -t flag in quotes.

PRKP-1013 "{0}: undefined environment variable for cluster database {1}"

Cause: The named environment variable is not defined for the named cluster database

Action: Set a value for the variable with "srvctl set env"

PRKP-1014 "{0}: undefined environment variable for instance {1} of cluster database {2}"

Cause: The named environment variable is not defined for the given instance

Action: Set a value for the variable with "srvctl set env"

PRKP-1015 "{0}: undefined environment variable"

Cause: The named environment variable is not defined

Action: Set a value for the named environment variable with "srvctl set env"

PRKP-1016 "Database {0} already enabled"

Cause: An attempt was made to enable a database that is already enabled.

Action: No action required.

PRKP-1017 "Instance {0} already enabled."

Cause: An attempt was made to enable an instance that is already enabled.

Action: No action required.

PRKP-1018 "Service {0} already enabled."

Cause: An attempt was made to enable a service that is already enabled.

Action: No action required.

PRKP-1019 "Database {0} already disabled."

Cause: An attempt was made to disable a database that is already disabled.

Action: No action required.

PRKP-1020 "Instance {0} already disabled."

Cause: An attempt was made to disable an instance that is already disabled.

Action: No action required.

PRKP-1021 "Service {0} already disabled."

Cause: An attempt was made to disable a service that is already disabled.

Action: No action required.

PRKP-1022 "The database {0} is still running."

Cause: An attempt was made to delete a database that is still running.

Action: Stop the database using 'srvctl stop database' before deleting the database.

PRKP-1023 "The instance {0} is still running."

Cause: An attempt was made to delete an instance that is still running.

Action: Stop the instance using 'srvctl stop instance' before deleting the instance.

PRKP-1024 "The service {0} is still running."

Cause: An attempt was made to delete a service that is still running.

Action: Stop the service using 'srvctl stop service' before deleting the service.

PRKP-1025 "The service {0} does not exist."

Cause: An attempt was made to operate on a non-configured service.

Action: Check if the service is configured through 'srvctl status service'.

PRKP-1026 "No instance found for database {0}."

Cause: An attempt was made to operate on a non-configured instance.

Action:

PRKP-1027 "Instance {0} is not found for database {1}."

Cause: An attempt was made to operate on a non-configured instance.

Action: Check if the instance is configured through 'srvctl config instance'.

PRKP-1028 "No preferred instance(s) for service {0}."

Cause: An attempt was made to create a service without preferred instances.

Action: Supply preferred instances for the service through 'srvctl create service'.

PRKP-1029 "Failed to register the service {0}."

Cause: Internal error.

Action: Contact support.

PRKP-1030 "Failed to start the service {0}."

Cause: Internal error.

Action: Contact support.

PRKP-1031 "Failed to stop the service {0}."

Cause: Internal error.

Action: Contact support.

PRKP-1032 "Cannot start the disabled service {0}."

Cause: Internal error.

Action: Contact support.

PRKP-1033 "Cannot relocate service {0} from instance {1} to instance {2}."

Cause: Internal error.

Action: Contact support.

PRKP-1034 "{0}: undefined environment variable for node {1}."

Cause: Internal error.

Action: Contact support.

PRKP-1035 "Invalid environment variable {0} setting for node {1}."

Cause:

Action: Contact support.

PRKP-1036 "Failed to unregister HA resource {0}."

Cause:

Action: Contact support.

PRKP-1037 "Failed to create cluster database {0}."

Cause:

Action: Contact support.

PRKP-1038 "Invalid instance {0} specified for the Service {1}."

Cause:

Action: Contact support.

PRKP-1039 "operation result is null"

Cause:

Action: Contact support.

PRKP-1044 "Failed to enable the database {0}."

Cause:

Action: Contact support.

PRKP-1045 "Failed to disable the database {0}."

Cause:

Action: Contact support.

PRKP-1046 "Failed to enable the instance {0}."

Cause: Cause

Action: Contact support.

PRKP-1047 "Failed to disable the instance {0}."

Cause:

Action: Contact support.

PRKP-1048 "Failed to change configuration for service {0}."

Cause:

Action: Contact support.

PRKP-1049 "{0}: undefined environment variable for service {1} of cluster database {2}"

Cause: The environment variable is not defined for the service.

Action: No action required.

PRKP-1050 "Failed to remove the service {0}."

Cause: There was a problem while executing the crs_unregister command.

Action: Verify whether the crs_unregister command succeeds in unregistering a CRS resource.

PRKP-1051 "Failed to remove the service {0} on instance {1}."

Cause: There was a problem while executing the crs_unregister command.

Action: Verify whether the crs_unregister command succeeds in unregistering a CRS resource.

PRKP-1052 "Failed to enable the service {0}."

Cause: There was a problem while executing the crs_register command.

Action: Verify whether the crs_register command succeeds in registering a CRS resource.

PRKP-1053 "Failed to disable the service {0}."

Cause: There was a problem while executing the crs_register command.

Action: Verify whether the crs_register command succeeds in registering a CRS resource.

PRKP-1054 "Failed to enable the service {0} on instance {1}."

Cause: There was a problem while executing the crs_register command.

Action: Verify whether the crs_register command succeeds in registering a CRS resource.

PRKP-1055 "Failed to disable the service {0} on instance {1}."

Cause: There was a problem while executing the crs_register command.

Action: Verify whether the crs_register command succeeds in registering a CRS resource.

PRKP-1056 "Failed to get the status of the resource {0}."

Cause: There was a problem while executing the crs_stat command.

Action: Verify whether the crs_stat command gives the status of the CRS resources registered.

PRKP-1057 "Failed to set the environment for service {0}."

Cause: There was a problem while accessing the OCR configuration registry.

Action: Check whether the OCR registry is accessible by executing a srvctl config command.

PRKP-1058 "Failed to unset the environment for service {0}."

Cause: There was a problem while accessing the OCR configuration registry.

Action: Check whether the OCR registry is accessible by executing a srvctl config command.

PRKP-1059 "Failed to get the environment for service {0}."

Cause: There was a problem while accessing the OCR configuration registry.

Action: Check whether the OCR registry is accessible by executing a srvctl config command.

PRKP-1060 "Failed to get CRS home."

Cause: Internal error.

Action: Contact support.

PRKP-1061 "Failed to modify the database {0}."

Cause: Internal error.

Action: Contact support.

PRKP-1062 "Service {0} is already running."

Cause: Attempt to start a service that is already running.

Action: None required.

PRKP-1063 "Service {0} is already stopped."

Cause: Attempt to stop a service that is already stopped.

Action: None required.

PRKP-1064 "Service {0} is already running on instance {1}."

Cause: Attempt to start a service on an instance where it is already running.

Action: None required.

PRKP-1065 "Service {0} is already stopped on instance {1}."

Cause: Attempt to stop a service on an instance where it is already stopped.

Action: None required.

PRKP-1066 "Instance {0} is not an available instance for service {1}."

Cause:

Action: Contact support.

PRKP-1067 "Instance {0} is the last available instance for service {1}. Try modify service instead."

Cause:

Action: Contact support

PRKP-1068 "Cannot stop the critical instance {0} in critical standby database {1} because it would result in a shutdown of the primary database."

Cause: Attempt to stop the critical instance in critical standby database while the primary database is running.

Action: Do not stop the critical instance in critical standby database while the primary database is running.

PRKP-1069 "Failed to change domain of the database {0} to {1}, because this domain name is already used by service {2} configured under the database."

Cause: Attempt to change the database domain when there are services configured with this domain.

Action: Do not change the database domain when there are services configured with that domain.

PRKP-1070 "Service name {0} contains illegal characters."

Cause: Invalid characters have been specified in the service name given.

Action: Supply a name for the service with the character set [a-zA-Z0-9_].

PRKP-1071 "Service Provider name or instance name {0} contains illegal characters."

Cause: Invalid characters have been specified in the service provider or instance name given.

Action: Supply a name for the service provider or instance with the character set [a-zA-Z0-9_].

PRKP-1072 "Failed to create service {0} for database {1}, because the specified service domain name is the same as the database domain {2}."

Cause:

Action:

PRKP-1073 "Cannot create database {0} because a database named {1} already exists."

Cause: Attempt to create a database that already exists.

Action: Choose a different name for the database being created.

PRKP-1074 "Failed to relocate a service resource to instance {0} during modifying service configuration for service {1}."

Cause: Internal error.

Action: Contact support.

PRKP-1075 "Instance {0} is the last preferred instance for service {1}."

Cause: Internal error.

Action: Contact support

PRKR—Cluster Registry Messages

PRKR-1001 "cluster database {0} does not exist"

Cause: The cluster database was never configured in the cluster registry.

Action: Check if the database has been configured by printing a list of all cluster databases using 'srvctl config'.

PRKR-1002 "cluster database {0} already exists"

Cause: An attempt was made to configure a cluster database which was already existing in the cluster registry.

Action: Check if the database has already been configured by printing a list of all cluster databases using 'srvctl config'.

PRKR-1003 "instance {0} does not exist"

Cause: The named instance is not configured in the cluster registry.

Action: Use srvctl options to check if the instance was configured in the cluster registry.

PRKR-1004 "instance {0} already exists"

Cause: The named instance is already configured in the cluster registry.

Action: Use srvctl options to check if the instance has already been configured in the cluster registry.

PRKR-1005 "adding of cluster database {0} configuration failed, {1}"

Cause: An error occurred while attempting to add cluster database configuration to the cluster registry.

Action: Check if the cluster registry is accessible by using 'ocrcheck' or 'srvctl config'

PRKR-1006 "deleting of cluster database {0} configuration failed, {1}"

Cause: Same as PRKR-1005.

Action: See earlier error messages.

PRKR-1007 "getting of cluster database {0} configuration failed, {1}"

Cause: Same as PRKR-1005.

Action: See earlier error messages.

PRKR-1008 "adding of instance {0} on node {1} to cluster database {2} failed, {3}"

Cause: Same as PRKR-1005.

Action: See earlier error messages.

PRKR-1009 "deleting of instance {0} from cluster database {1} failed, {2}"

Cause: Same as PRKR-1005.

Action: See earlier error messages.

PRKR-1010 "moving of instance {0} to node {1} of cluster database {2} failed, {3}"

- Cause:** Same as PRKR-1005.
Action: See earlier error messages.
- PRKR-1011 "renaming of instance {0} to instance {1} of cluster database {2} failed, {3}"**
Cause: Same as PRKR-1005.
Action: See earlier error messages.
- PRKR-1016 "reading of cluster database {0} configuration failed, {1}, {2}"**
Cause: Same as PRKR-1005.
Action: See earlier error messages.
- PRKR-1017 "writing of cluster database {0} configuration failed, {1}, {2}"**
Cause: Same as PRKR-1005.
Action: See earlier error messages.
- PRKR-1018 "reading of directory failed, {0}, {1}"**
Cause: Internal error.
Action: Contact your customer support representative.
- PRKR-1019 "writing of directory failed, {0}, {1}"**
Cause: Internal error.
Action: Contact your customer support representative.
- PRKR-1020 "reading of version information failed, {0}, {1}"**
Cause: Same as PRKR-1005.
Action: See earlier error messages.
- PRKR-1021 "writing of version information failed, {0}, {1}"**
Cause: Same as PRKR-1005.
Action: See earlier error messages.
- PRKR-1022 "raw device {0} contains incompatible version, {1} != {2}"**
Cause: An attempt was made to use an incompatible version of the cluster registry.
Action: Contact your customer support representative.
- PRKR-1023 "file {0} does not exist"**
Cause: The named file did not exist.
Action: Check if the file exists.
- PRKR-1024 "file {0} does not have {1} permissions"**
Cause: The named file did not have the specified permission.
Action: Try changing the permission on the file to the specified permission.
- PRKR-1025 "file {0} does not contain property {1}"**
Cause: The file did not contain the specified property.
Action: Contact your customer support representative.
- PRKR-1026 "property {0} not set in file {1}"**
Cause: The file did not contain the specified property.
Action: Contact your customer support representative.

PRKR-1027 "failed to retrieve list of cluster databases"

Cause: Same as PRKR-1005.

Action: See earlier error messages.

PRKR-1028 "raw device {0} is invalid\n[HINT: initialize raw device by using \"srvconfig\" tool]"

Cause: The cluster registry was never initialized.

Action: Use srvconfig -init to initialize the cluster registry.

PRKR-1038 "invalid argument {0} specified to -init option"

Cause:

Action:

PRKR-1039 "invalid option {0} specified"

Cause: The specified option was invalid.

Action: Check usage.

PRKR-1040 "missing <file> argument for {0} option"

Cause: The specified option was invalid for srvconfig

Action: Check usage for details.

PRKR-1045 "raw device version \"{0}\""

Cause: Attempted to retrieve the version of the cluster registry.

Action: No action is required.

PRKR-1046 "srvconfig detected valid raw device \"{0}\" \n[HINT: please specify -init -f option to forcefully initialize it]"

Cause: A valid cluster registry was detected.

Action: No action is required.

PRKR-1047 "raw device {0} is in use by daemon(s) on node(s) {1}"

Cause: An attempt was made to initialize the cluster registry while the Global Services Daemons were up on one or more nodes in the cluster.

Action: Stop all Global Services Daemons on all nodes in the cluster by using 'gsdctl stop' command on every node. Try the 'srvconfig -init' operation again.

PRKR-1050 "file {0} creation in {1} directory failed, check permissions, etc."

Cause: Attempted to create a file in a directory which did not exist or which did not have the right permissions.

Action: Create the directory if it did not exist or change the permission of the directory.

PRKR-1051 "file {0} does not contain an entry for dbname {1}"

Cause: Internal error.

Action: Contact your customer support representative.

PRKR-1052 "file name {0} is not of <cluster database name>.conf form"

Cause: An attempt was made to register a cluster database in the cluster registry and the file argument passed was not of <cluster database name>.conf form.

Action: See usage of srvconfig for more information.

PRKR-1053 "invalid range {0} specified in node_list = {1}"

Cause:**Action:** Contact your customer support representative.**PRKR-1054 "invalid parameter {0} specified in inst_oracle_sid = {1}"****Cause:** Extra number of arguments provided to srvconfig.**Action:** See usage of srvconfig for details.**PRKR-1055 "invalid extra arguments {0} specified to {1} option"****Cause:** Provided invalid arguments to srvconfig.**Action:** See usage of srvconfig for details.**PRKR-1056 "invalid registry entry {0} found, should be of form {1}"****Cause:** Detected an invalid registry entry while attempting to add a 8.1.7 or earlier version of Oracle Parallel Server in the cluster registry.**Action:** Contact your customer support representative.**PRKR-1057 "environment variable does not exist"****Cause:** Attempted to retrieve non existing environment variable.**Action:** Set the environment variable.**PRKR-1058 "Service {0} does not exist in cluster database {1}."****Cause:** The named service is not configured in the cluster registry.**Action:** Use srvctl options to check if the service was configured in the cluster registry.**PRKR-1059 "Node {0} does not exist."****Cause:** Node applications for the named node are not configured in the cluster registry.**Action:** Use srvctl options to check if the node applications were configured for the given node in the cluster registry.**PRKR-1060 "Failed to add configuration for node {0}"****Cause:****Action:** Contact your customer support representative.**PRKR-1061 "Failed to run remote command to get node configuration for node {0}"****Cause:** An internal error occurred while retrieving configuration for the given node.**Action:** Contact your customer support representative.**PRKR-1062 "Failed to find configuration for node {0}"****Cause:****Action:** Contact your customer support representative.**PRKR-1063 "VIP {0} is already existing"****Cause:****Action:** Contact your customer support representative.**PRKR-1064 "SRVM configuration operation failed due to Oracle Cluster Registry error :"****Cause:** Error occurred while accessing the Oracle Cluster Registry.

Action: Contact your customer support representative.

PRKR-1066 "cluster database domain does not match"

Cause:

Action: Contact your customer support representative.

PRKR-1067 "Failed to get environment for cluster database {0}, {1}"

Cause: Unable to retrieve the environment configuration for the given cluster database from the cluster registry.

Action: Use ocrdump to check if environment was configured for the given cluster database in the cluster registry.

PRKR-1068 "Failed to get environment for instance {1} of cluster database {0}, {2}"

Cause: Unable to retrieve the environment configuration for the given instance from the cluster registry.

Action: Use ocrdump to check if environment was configured for the given instance in the cluster registry.

PRKR-1069 "Failed to set environment for cluster database {0}, {1}"

Cause: Unable to update the environment configuration for the given cluster database to the cluster registry.

Action: Contact your customer support representative.

PRKR-1070 "Failed to set environment for instance {1} of cluster database {0}, {2}"

Cause: Unable to update the environment configuration for the given instance to the cluster registry.

Action: Contact your customer support representative.

PRKR-1071 "Failed to unset environment for cluster database {0}, {1}"

Cause: Unable to update the environment configuration for the given cluster database to the cluster registry.

Action: Contact your customer support representative.

PRKR-1072 "Failed to unset environment for instance {1} of cluster database {0}, {2}"

Cause: Unable to update the environment configuration for the given instance to the cluster registry.

Action: Contact your customer support representative.

PRKR-1073 "\n##### Configuration of nodeapps follows #####\n"

Cause:

Action: Contact your customer support representative.

PRKR-1074 "\n##### Configuration of vip_range follows #####\n"

Cause:

Action: Contact your customer support representative.

PRKR-1075 "Insufficient privileges for doing this operation"

Cause: User did not have sufficient privileges when running this command.

Action: Execute this command as a privileged user.

PRKR-1076 "This command cannot run when the RAC daemons (crsd, evmd, ocssd) are running. Make sure the daemons are not running before invoking this command"

Cause: The RAC daemons were running when this command was invoked.

Action: Make sure that the RAC daemons have been stopped before running this command.

PRKR-1077 "One or more arguments passed to the function are not valid"

Cause: One or more invalid arguments were passed to the given method.

Action: Contact your customer support representative.

PRKR-1078 "Database {0} cannot be administered using current version of srvctl. Instead run srvctl from {1}"

Cause: Using incorrect SRVCTL version.

Action: Contact your customer support representative.

PRKR-1079 "Failed to initialize the Oracle Cluster Registry"

Cause: Failed to initialize the Oracle Cluster Registry.

Action: Contact your customer support representative.

PRKS—Automatic Storage Management Messages

PRKS-1000: "ASM instance "{0}" already exists on node "{1}"

Cause: An attempt was made to add configuration for ASM instance on the node where it already exists.

Action: Check if ASM instance was configured on the node using 'srvctl config asm -n <node>' before adding configuration for it.

PRKS-1001: ASM instance "{0}" does not exist on node "{1}"

Cause: The configuration for ASM instance doesn't exist on the node.

Action: Check if ASM instance was configured on the node using 'srvctl config asm -n <node>' before performing the operation.

PRKS-1002: Failed to create CRS profile for ASM instance "{0}" on node "{1}", [{2}]

Cause: 'crs_stat -p' command failed for the ASM instance resource on the node.

Action: Contact Oracle Support.

PRKS-1003: Failed to register CRS resource for ASM instance "{0}" on node "{1}", [{2}]

Cause: crs_register command failed for ASM instance resource on the node.

Action: Contact Oracle Support.

PRKS-1004: Failed to unregister CRS resource for ASM instance "{0}" on node "{1}", [{2}]" }

Cause: crs_unregister command failed for the ASM instance resource on the node.

Action: Check if there is a database instance dependent upon the ASM instance on the node using crs_stat -p ora.<db>.<inst>.inst and see if ASM instance resource name appears in the required resources for the database instance. Remove the database instance's required resource dependency on ASM instance using 'srvctl modify asm' command before retrying this operation.

PRKS-1005: Failed to create CRS resource for ASM instance "{0}" on node "{1}", [{2}]

Cause: crs_register command failed for the ASM instance resource on the node.

Action: Contact Oracle Support.

PRKS-1006: ASM instance "{0}" is already running on node "{1}".

Cause: An attempt was made to start a running ASM instance on the node.

Action: None.

PRKS-1007: ASM instance "{0}" is still running on node "{1}".

Cause: An attempt was made to remove the configuration for a running ASM instance on the node.

Action: Stop the ASM instance using 'srvctl stop asm -n <node> -i <inst>' command before performing the remove operation.

PRKS-1008: ASM instance "{0}" is not running on node "{1}".

Cause: An attempt was made to stop a non-running ASM instance on the node.

Action: None.

PRKS-1009: Failed to start ASM instance "{0}" on node "{1}", [{2}]

Cause: crs_start failed for the ASM instance resource on the node, may be due to invalid startup credentials or missing parameter file.

Action: Check if VIP resource is online on the node and then try to startup the ASM instance using SQL*Plus to get more diagnostic information.

PRKS-1010: Failed to start ASM instances "{0}" on node "{1}", [{2}]

Cause: See above.

Action: See above.

PRKS-1011: Failed to check status of ASM instance "{0}" on node "{1}", [{2}]

Cause: crs_stat failed for the ASM instance resource on the node.

Action: Contact Oracle Support.

PRKS-1012: Failed to stop ASM instance "{0}" on node "{1}", [{2}]"

Cause: crs_stop failed for the ASM instances on the node, may be due to CRS resource dependency or invalid stop ASM instance credentials.

Action: Check if there are database instances dependent upon the ASM instance on the node using crs_stat -p ora.<db>.<inst>.inst and see if ASM instance resource names appear in the required resources for the database instance. Stop the database instances first using 'srvctl stop instance' command before retrying ASM instances stop operation or try to stop the ASM instance using SQL*Plus to get more diagnostic information.

PRKS-1013: Failed to stop ASM instances "{0}" on node "{1}", [{2}]"

Cause: See above.

Action: See above.

PRKS-1014: Failed to enable CRS resource for ASM instance "{0}" on node "{1}", [{2}]

Cause: Failed to set the enable flag of ASM instance configuration in cluster registry.

Action: Check if cluster registry is accessible by executing 'srvctl config ' command.

PRKS-1015: Failed to enable CRS resources for ASM instances "{0}" on node "{1}", [{2}]"

Cause: See above for each ASM instance.

Action: Check if cluster registry is accessible by executing 'srvctl config ' command.

PRKS-1016: Failed to disable CRS resource for ASM instance "{0}" on node "{1}", [{2}]"

Cause: Failed to reset enable flag of ASM instance configuration in cluster registry.

Action: Check if cluster registry is accessible by executing 'srvctl config ' command.

PRKS-1017: Failed to disable CRS resources for ASM instances "{0}" on node "{1}", [{2}]"

Cause: See above for each ASM instance.

Action: Check if cluster registry is accessible by executing 'srvctl config ' command.

PRKS-1019: Cannot create CRS dependency between database instance "{0}" configured on node "{1}" and ASM instance "{2}" configured on node "{3}".

Cause: An attempt was made to create CRS dependency between database instance and ASM instance configured on different nodes.

Action: Make sure that database instance and ASM instance are configured on the same node for creating CRS dependency between them.

PRKS-1020: Failed to create CRS dependency between database instance "{0}" and ASM instance "{1}", [{2}]

Cause: 'crs_register -u' failed to create CRS dependency between the database instance and the ASM instance on the node.

Action: Contact Oracle support.

PRKS-1021: Failed to remove CRS dependency between database instance "{0}" and ASM instance "{1}", [{2}]

Cause: 'crs_register -u' command failed to remove CRS dependency between the database instance and the ASM instance on the node.

Action: Contact Oracle Support.

PRKS-1022: Failed to remove CRS dependency between database instance "{0}" and ASM instances "{1}", [{2}]

Cause: See above.

Action: See above.

PRKS-1023: Failed to remove CRS resource for ASM instance "{0}" on node "{1}", [{2}]

Cause: crs_unregister command failed to unregister CRS resource for the ASM instances on the node.

Action: Contact Oracle Support.

PRKS-1026: ASM Configuration for node "{0}" does not exist in cluster registry."

Cause: An attempt was made to retrieve names of ASM instances configured on the node without configuring any ASM instance on it.

Action: First configure ASM instance using 'srvctl add asm' on the node command before executing the get configuration operation.

PRKS-1027: Configuration for ASM instance "{0}" does not exist in cluster registry."

Cause: An attempt was made to retrieve node name for a given ASM instance name which is not configured on any cluster nodes.

Action: First configure ASM instance using 'srvctl add asm' command before executing the get configuration operation.

PRKS-1028: Configuration for ASM instance "{0}" on node "{1}" does not exist in cluster registry.

Cause: An attempt was made to retrieve configuration for the ASM instance on the node name where it was not configured.

Action: Use 'srvctl config asm -n <node>' command to determine the ASM instance names configured on it and then pass one of these names as '-i <inst>' argument to 'srvctl config asm' command.

PRKS-1029: Client version "{0}" is not compatible with ASM instance configuration version "{1}" in cluster registry.

Cause: Version of the client that tried to retrieve ASM instance configuration is not compatible with ASM instance configuration version.

Action: Make sure client version is compatible with ASM instance configuration version before accessing it.

PRKS-1030: Failed to add configuration for ASM instance "{0}" on node "{1}" in cluster registry, [{2}]

Cause: Failed to add configuration for the ASM instance in cluster registry.

Action: Check if cluster registry is accessible by executing 'srvctl config' command.

PRKS-1031: "Failed to retrieve configuration for ASM instance "{0}" on node "{1}" from cluster registry, [{2}]

Cause: Failed to retrieve configuration for the ASM instance from cluster registry.

Action: Check if cluster registry is accessible by executing 'srvctl config' command.

PRKS-1032: Failed to modify configuration for ASM instance "{0}" on node "{1}" in cluster registry, [{2}]

Cause: Failed to modify configuration for the ASM instance on the node in cluster registry.

Action: Check if cluster registry is accessible by executing 'srvctl config' command.

PRKS-1033: Failed to remove configuration for ASM instance "{0}" on node "{1}" from cluster registry, [{2}]

Cause: Failed to remove configuration for the ASM instance from cluster registry.

Action: Check if cluster registry is accessible by executing 'srvctl config'

PRKS-1034: Failed to remove configuration for ASM instances "{0}" on node "{1}" from cluster registry, [{2}]"

Cause: See above.

Action: See above.

- PRKS-1035: Failed to retrieve ORACLE_HOME value for ASM instance "{0}" on node "{1}" from cluster registry, [{2}]**
Cause: Failed to retrieve ORACLE_HOME from the ASM instance configuration in cluster registry.
Action: Check if cluster registry is accessible by executing 'srvctl config'
- PRKS-1036: VIP resource for ASM instance node "{0}" does not exist."**
Cause: VIP resource for the node doesn't exist, it a required resource for ASM instance configuration on the node.
Action: Configure the VIP resource for the node using 'srvctl add nodeapps' command as privileged user before adding configuration for ASM instance.
- PRKS-1037: Failed to check existence of VIP resource for ASM instance node "{0}", [{1}]**
Cause: 'crs_stat ora.<node>.vip' failed to check status of the VIP resource for the node.
Action: Contact Oracle Support.

PRKU—Command Line Parser Utility Messages

- PRKU-1000: "Exception Caused by: "**
Cause: The current exception was caused by another earlier exception.
Action: Examine all of the nested exceptions to determine the root cause of the error.
- PRKU-1001: "The parameter {0} is required."**
Cause: The parameter specified is required by a command line utility.
Action: Supply the parameter specified.
- PRKU-1002: "Unexpected argument {0}."**
Cause: The argument specified is not needed or not understood by this utility.
Action: Verify the arguments on the command you are executing.
- PRKU-1003: "The parameter {0} requires an argument."**
Cause: The parameter specified must be accompanied by an argument.
Action: Verify the arguments on the command you are executing.

PRKV — Virtual IP Configuration Assistant Messages

- PRKV-1013 "A network interface is required"**
Cause: A network interface is not provided for configuring virtual IP address.
Action: when running VIPCA in GUI mode, select a suitable network interface from the list. When running VIPCA in silent mode, provide a suitable interface using '-interfaces' option.
- PRKV-1014 "Enter a valid IP address for the node \"{0}\"."**
Cause: An IP address to be configured as virtual IP is not entered for the node.
Action: Enter an unused IP address to be used for configuring virtual IP address.
- PRKV-1015 "Enter a valid subnet mask for the IP address \"{0}\"."**
Cause: An IP netmask for the IP address is not entered.

Action: Enter netmask for the IP address.

PRKV-1016 "IP address \"{0}\" is invalid. Enter a valid IP address."

Cause: IP address entered is invalid.

Action: Enter a valid IP address in standard IEEE format.

PRKV-1017 "IP address \"{0}\" does not exist. Enter a valid IP address."

Cause: IP address cannot be resolved in the network.

Action: Add an IP address in DNS or /etc/hosts so it can be resolved.

PRKV-1018 "IP address \"{0}\" has invalid format. Enter a valid IP address."

Cause: IP address is not in standard IEEE format.

Action: Enter a valid IP address in IEEE format.

PRKV-1019 "Netmask entered \"{0}\" has invalid format. Enter a valid netmask."

Cause: Netmask entered in not in standard IEEE format.

Action: Enter a valid IP netmask in IEEE format.

PRKV-1039 "IP address \"{0}\" has already been used. Enter an unused IP address."

Cause: IP address entered is already used by another node as virtual IP address.

Action: Enter an unused IP address.

PRKV-1060 "Invalid node name \"{0}\" entered in an input argument."

Cause: An attempt has been made to configure CRS node applications for the node that is not part of the cluster.

Action: Check if the node is configured in the cluster using '<CRS home>/bin/olsnodes' Enter only the nodes that are part of the cluster, or add the node to the cluster before configuring nodeapps.

PRKV-1061 "Invalid IP address \"{0}\" entered in an input argument."

Cause: The following may be a cause of this error: a) IP address is not in valid IEEE format, b) IP address cannot be resolved in the network, or c) IP address entered is already being used.

Action: Check the format of IP address, check if it exists in network DNS or /etc/hosts, or check if it already been configured to use.

PRKV-1062 "Invalid netmask \"{0}\" entered in an invalid argument."

Cause: IP netmask entered is not in valid IEEE format,

Action: Enter IP netmask in IEEE format.

PRKV-1063 "Invalid interface \"{0}\" entered in an input argument."

Cause: Interface entered as an input are not found on the system.

Action: Check the list of interfaces configured to the system.

PRKV-1064 "Insufficient privileges."

Cause: You do not have Administrative privileges to perform operation.

Action: Login as Administrator and run the tool again.

PRKV-1065 "Failed to obtain handle to CRS."

Cause: An attempt has been made to connect to Oracle Cluster Ready Service while it is not up and running.

Action: Check the status Oracle Cluster Ready Services.

PRKV-1066 "Check the log file \"{0}\" and fix the problem before retrying."

Cause: There is a problem in starting the resource. The exact nature of the problem can be found using the trc file shown. Resource cannot start unless the problem is fixed.

Action: Fix the problem and retry the operation.

PRKV-1067 "Check the log file \"{0}\" for more details"

Cause: The tool could not create or start the resource. Check the specified file for details of the nature of the problem.

Action: Fix the problem first to continue.

**PRKV-1068 "Host IP address \"{0}\" cannot be used as virtual IP for the node \"{1}\"
Enter a different IP address."**

Cause: An attempt has been made to configure the host as virtual IP that can failover to another node.

Action: Enter an unused IP address that different from host IP address.

**PRKV-1069 "Host IP address \"{0}\" of cluster node \"{1}\" cannot be used as virtual
IP for the node \"{2}\". Enter a different IP address."**

Cause: An attempt has been made to configure the remote host as virtual IP that can failover.

Action: Enter an unused IP address that different from host IP address of any node in the cluster.

Index

A

administering instances
 with Server Management, 8-2

affinity
 awareness, 7-3

alert log
 managing, A-2

alert logs, A-5

ALTER DATABASE statement
 CLOSE clause, 8-6

ALTER SYSTEM ARCHIVE LOG CURRENT
 statement, 8-6

ALTER SYSTEM ARCHIVE LOG statement, 8-6
 INSTANCE option, 8-6

ALTER SYSTEM CHECKPOINT LOCAL
 statement, 8-6

ALTER SYSTEM CHECKPOINT statement
 global versus local, 8-6
 specifying an instance, 8-6

ALTER SYSTEM SET CHECKPOINT LOCAL
 statement, 8-6

ALTER SYSTEM SWITCH LOGFILE statement, 8-6

ARCHIVE LOG command, 8-6

archive logs
 file format and destination, 6-4
 file naming, 6-3

archiving redo log files
 identified in control file, 3-4
 log sequence number, 6-4

ASM
 see Automatic Storage Management

asm
 as SRVCTL noun, B-5

Automatic Storage Management, 3-2

Automatic Storage Management instance, 3-2

B

background processes
 SMON, 2-4, 7-2

background thread trace files, A-2

BACKGROUND_DUMP_DEST parameter, A-2, A-5

backups
 server parameter file, 2-10

blocks

 associated with instance, 7-2

buffer cache
 instance recovery, 7-2

C

Cluster Database Management Error Messages
 (PRKP), C-15

cluster file system
 storage in Real Application Clusters, 3-2

Cluster Manager (CSS)
 log files, A-3

Cluster Ready Services (CRS), 3-5

comments
 specifying in server parameter file, 2-7

committed data
 instance failure, 7-2

configuration problems
 reinitializing the OCR, 3-7

CONNECT command, 2-2, 8-6

CONNECT SYS
 example of, 2-4

connecting
 to instances, 2-2

control files
 MAXLOGHISTORY, 3-4

CREATE DATABASE statement
 MAXLOGHISTORY clause, 3-4

creating
 server parameter files, 2-5, 2-10

D

database
 as SRVCTL noun, B-5
 number of archived log files, 3-4
 quiescing, 8-7

Database Configuration Assistant
 Database Storage page, 5-10
 Instance Management page, 5-10
 List of Cluster Database Instances page, 5-10
 List of Cluster Databases page, 5-10
 Operations page, 5-10
 Summary dialog, 5-11
 Welcome page, 5-10

Database Storage page, 5-10

DISCONNECT command, 2-2
disconnecting from instances, 2-2

E

Enterprise Manager, 2-2, 8-2
error messages
 for management tools, C-1
Event Manager (EVM)
 log files, A-3

F

failure
 instance, 7-2
 multiple node, 7-2
 node, 7-2
features, new, 1-xix
files
 archiving redo log, 6-4
 control file, 3-4
 redo log, 6-4

G

GLOBAL clause
 forcing a checkpoint, 8-6

H

HOST command, 8-6

I

initdb_name.ora file
 BACKGROUND_DUMP_DEST parameter, A-2,
 A-5
 USER_DUMP_DEST parameter, A-2
initialization parameters
 cluster database issues regarding, 2-8
 RECOVERY_PARALLELISM, 7-8
 that must be identical on all instances, 2-7
 that must be unique on all instances, 2-8
instance
 as SRVCTL noun, B-5
Instance Management page, 5-10
INSTANCE option, 8-6
INSTANCE_NUMBER parameter, 2-8
instances
 failure, 7-2
 failures, recovery from, 7-2
 recovery, 2-4, 7-2
 recovery, abnormal shutdown, 2-4
 recovery, multiple failures, 7-2
 Server Management, 8-2
 shutting down, 2-3
 verifying, 8-6

J

java-based tools

tracing, A-4

L

List of Cluster Database Instances page, 5-10
List of Cluster Databases page, 5-10
LOCAL clause
 forcing a checkpoint, 8-6
log files
 for CRS, A-2
 for CSS, A-3
 for EVM, A-3
 for OCR, A-3
log sequence numbers, 6-4
log switches, 3-4
LOG_ARCHIVE_FORMAT parameter, 6-4

M

MAXLOGHISTORY option, 3-4
media failures
 recovery from, 7-6
messages
 errors for management tools, C-1
modified data
 instance recovery, 7-2
multiple node failures, 7-2
multiple nodes
 starting from one node, 2-4
multiplexed redo log files, 3-4

N

network interfaces
 defining with OIFCFG, 8-8
new features, 1-xix
nodeapps
 as SRVCTL noun, B-5
nodes
 affinity awareness, 7-3
 failure of, 7-2

O

OCR
 clients on UNIX, stopping, 3-6
ocrconfig -export, 3-8
OIFCFG, 8-7
 commands, 8-8, 8-9
 defining network interfaces, 8-8
 if_name element, 8-8
 if_type element, 8-8
 nodename element, 8-8
 subnet element, 8-8
 syntax, 8-8
online recovery, 7-2
operating system-specific Oracle documentation
 archived redo log name, 6-4
Operations page, 5-10
Oracle Cluster Registry (OCR), 3-5
 administering, 3-5

- log files, A-3
- managing, 3-5
- Oracle Interface Configuration tool
 - see OIFCFG
- Oracle Real Application Clusters
 - overview of administration, 1-1
- ORACLE_SID parameter, 2-8
- oraxxxx.trc file, A-2

P

- parallel recovery, 7-9
- PARALLEL_MAX_SERVERS parameter, 7-9
- parameter file
 - overview, 2-5
- parameter files
 - server file, exporting, 2-7
- parameters
 - that must be identical on all instances, 2-7
 - that must be unique on all instances, 2-8
- PRKC messages, C-2
- PRKP
 - messages, C-15
- PRKR
 - messages, C-22

Q

- quiesce database
 - in Real Application Clusters, 8-7

R

- RECOVER command, 7-9, 8-6
- recovery
 - after SHUTDOWN ABORT, 2-4
 - from multiple node failure, 7-2
 - from single-node failure, 7-2
 - instance, 2-4
 - media failures, 7-6
 - online, 7-2
 - parallel, 7-9
 - PARALLEL_MAX_SERVERS parameter, 7-9
- RECOVERY_PARALLELISM parameter, 7-8
- redo log files
 - identified in control file, 3-4
 - instance recovery, 7-2
 - log sequence number, 6-4
 - using, 3-4
- redo log groups, 3-4
- redo logs
 - format and destination specifications, 6-4
- Repository Error Messages (PRKR), C-22
- resources
 - releasing, 7-2
- rolling back
 - instance recovery, 7-2

S

- scalability

- adding nodes and instances, 5-2
- sequences
 - log sequence number, 6-4
- Server Management
 - administration of instances, 8-2
- server parameter file
 - backing up, 2-10
 - exporting, 2-7
 - setting values in, 2-6
- server parameter files
 - creating, 2-5, 2-10
- service
 - as SRVCTL noun, B-5
- SERVICE_NAMES
 - parameter, 4-3
- services, 4-2
 - adding, modifying, 4-2
 - names, limitations for, 4-3
- sessions
 - multiple, 2-2
- setting instances, 2-2
- SHOW INSTANCE command, 8-6
- SHOW PARAMETER command, 8-6
- SHOW PARAMETERS command, 8-6
- SHOW SGA command, 8-6
- SHUTDOWN ABORT command, 2-3
- SHUTDOWN command, 8-6
 - ABORT option, 2-4
- SHUTDOWN TRANSACTIONAL, 2-4
- shutting down
 - instances, 2-3
- shutting down an instance
 - abnormal shutdown, 2-4
- shutting down instances, 2-3
- sidalrt.log file, A-5
- siddbwr.trc file, A-2
- sidsmon.trc file, A-2
- SMON process
 - instance recovery, 7-2
 - recovery after SHUTDOWN ABORT, 2-4
- snapshot control file, 6-2
- SQL statements
 - instance-specific, 8-6
- SQL*Plus, 2-2, 8-5
- SRVCONFIG
 - importing and exporting SRVM information
 - with, 3-8
- SRVCTL
 - add asm command, B-8
 - add database command, B-6
 - add instance command, B-6
 - add nodeapps command, B-8
 - add service command, B-7
 - add, usage description, B-5
 - cluster database configuration tasks, B-3
 - cluster database tasks, B-3
 - concurrent commands, B-2
 - config asm command, B-10
 - config database command, B-9
 - config nodeapps command, B-10

- config service command, B-9
- config, usage description, B-8
- disable asm command, B-14
- disable database command, B-13
- disable instance command, B-13
- disable service command, 4-5, B-13
- disable, usage description, B-12
- enable asm command, B-12
- enable database command, B-11
- enable instance command, B-11
- enable service command, 4-5, B-11
- enable, usage description, B-10
- getenv database command, B-27
- getenv instance command, B-27
- getenv nodeapps command, B-28
- getenv service command, B-27
- getenv, usage description, B-26
- modify database command, B-20
- modify instance command, B-21
- modify nodeapps command, B-23
- modify service command, B-22
- modify, usage description, B-20
- node-level tasks, B-3
- overview, 2-3
- relocate service command, B-24
- relocate, usage description, B-23
- remove asm command, B-34
- remove database command, B-32
- remove instance command, B-32
- remove nodeapps command, B-33
- remove service command, B-33
- remove, usage description, B-32
- setenv database command, B-29
- setenv instance command, B-29
- setenv nodeapps command, B-30
- setenv service command, B-29
- setenv, usage description, B-28
- start asm command, B-16
- start database command, 2-5, B-15
- start instance command, 2-5, B-15
- start nodeapps command, B-16
- start service command, 4-5, B-16
- start, usage description, B-14
- status asm command, B-26
- status database command, B-24
- status instance command, B-25
- status nodeapps command, B-26
- status service command, 4-5, B-25
- status, usage description, B-24
- stop asm command, B-19
- stop database command, 2-5, B-17
- stop instance command, 2-5, B-18
- stop nodeapps command, B-19
- stop service command, B-18
- stop, usage description, B-17
- unsetenv database command, B-30
- unsetenv instance command, B-31
- unsetenv nodeapps command, B-31
- unsetenv service command, B-31
- unsetenv, usage description, B-28

- STARTUP command, 8-6
 - storage
 - cluster file system, 3-2
- Summary dialog, 5-11
- SYSDBA
 - privilege for connecting, 2-3
- SYSOPER privilege
 - for connecting, 2-3
- system change, 6-4

T

- THREAD parameter, 2-9
- trace files, A-2
 - background thread trace files, A-2
 - managing, A-2
 - oraxxxx.trc, A-2
 - sidalrt.log, A-5
 - siddbwr.trc file, A-2
 - sidsmon.trc file, A-2
- tracing
 - for CRS, A-4
 - java-based tools, A-4
- transactions
 - instance failure, 7-2
 - rolling back, 7-2
 - waiting for recovery, 7-2
- troubleshooting, A-2

U

- UNDO_TABLESPACE parameter, 2-8
- user trace files, A-2
- USER_DUMP_DEST parameter, A-2

V

- V\$ACTIVE_INSTANCES, 8-6
- verification
 - datafiles, online files, 3-4

W

- Welcome page, 5-10