# Oracle9*i* Real Application Clusters

Administration

Release 2 (9.2)

March 2002

Part No. A96596-01

Oracle9*i* Real Application Clusters Administration, Release 2 (9.2)

Part No. A96596-01

Primary Author: Mark Bauer.

Primary Contributors: Lance Ashdown, Jonathan Creighton, Raj Kumar, and Francisco Sanchez.

Contributors: David Austin, Jack Cai, Sohan Demel, Mitch Flatland, Carmen Frank, Vijay Lunawat, Dipak Saggi, Klaus Thielen, and Steve Wertheimer.

Graphic Designer: Valarie Moore.

# Contents

## 3 Administering Storage Components in Real Application Clusters

## 4 Administering Real Application Clusters Databases with the Server Control Utility, SQL, and SQL*Plus

## Part II Using Oracle Enterprise Manager to Administer Real Application Clusters

## 5 Administering Real Application Clusters Databases with Oracle Enterprise Manager

## Part III Backup and Recovery in Real Application Clusters

## 6 Using Recovery Manager in Real Application Clusters Environments

## 7 Performing Backup and Recovery in Real Application Clusters Environments

## Part IV Scalability in Real Application Clusters Environments

## 8 Adding Nodes and Instances and Deleting Instances in Real Application Clusters

## Part V Reference

## A Troubleshooting

## B Associating Instances and Users with Free Lists and Free List Groups (Optional)

## C Real Application Clusters Management Tools Error Messages

## Glossary

## Index

# List of Figures

## List of Tables

# Send Us Your Comments

**Oracle9*i* Real Application Clusters Administration, Release 2 (9.2)**

**Part No. A96596-01**

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. 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?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, 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  
  500 Oracle Parkway, Mailstop 4op11  
  Redwood Shores, CA 94065  
  USA

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

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

# Preface

*Oracle9i Real Application Clusters Administration* explains the **Real Application Clusters**-specific administrative tasks that supplement single **instance** administrative tasks. Information in this manual applies to Real Application Clusters as it runs on all operating systems. Where necessary, this manual refers to platform-specific documentation.

> **See Also:** The *Oracle9i Real Application Clusters Documentation Online Roadmap* to help you use the online Real Application Clusters documentation set

This preface contains these topics:

- Audience
- Organization
- Related Documentation
- Conventions
- Documentation Accessibility

## Audience

*Oracle9i Real Application Clusters Administration* is written primarily for network or Database Administrator responsible for the administration of Real Application Clusters.

To use this document you should first read the *Oracle9i Database Administrator's Guide* to become familiar with single-instance Oracle database administrative procedures. You should then read *Oracle9i Real Application Clusters Concepts* for a conceptual understanding of Real Application Clusters processing. You should also have installed Real Application Clusters using the document *Oracle9i Real Application Clusters Setup and Configuration* and any platform-specific documentation.

## Organization

This document contains the following five parts:

### Part I: "Introduction to Administering Real Application Clusters"

Part I describes Real Application Clusters administration and your initial administrative tasks.

### Chapter 1, "Introduction to Real Application Clusters Administration"

This chapter introduces the administrative tasks for Real Application Clusters software.

### Chapter 2, "Parameter Management in Real Application Clusters Environments"

This chapter describes parameter files and Real Application Clusters-specific parameters.

### Chapter 3, "Administering Storage Components in Real Application Clusters"

This chapter explains how to administer storage components in Real Application Clusters.

### Chapter 4, "Administering Real Application Clusters Databases with the Server Control Utility, SQL, and SQL*Plus"

This chapter explains how to administer Real Application Clusters databases with Server Control, SQL, and SQL*Plus.

### Part II: Using Oracle Enterprise Manager to Administer Real Application Clusters

Part II describes how to use Oracle Enterprise Manager to administer Real Application Clusters databases.

### Chapter 5, "Administering Real Application Clusters Databases with Oracle Enterprise Manager"

This chapter describes how to use Oracle Enterprise Manager to administer Real Application Clusters databases.

### Part III: Backup and Recovery in Real Application Clusters

Part III provides backup and recovery procedures for Real Application Clusters.

### Chapter 6, "Using Recovery Manager in Real Application Clusters Environments"

This chapter describes how to configure RMAN for Real Application Clusters.

### Chapter 7, "Performing Backup and Recovery in Real Application Clusters Environments"

This chapter explains how to backup and recover Real Application Clusters databases.

### Part IV: Scalability in Real Application Clusters Environments

Part IV provides information about adding nodes and instances to scale your Real Application Clusters environment.

### Chapter 8, "Adding Nodes and Instances and Deleting Instances in Real Application Clusters"

This chapter explains how to add nodes and instances and how to delete instances in Real Application Clusters using the Oracle Universal Installer and the Database Configuration Assistant.

### Part V: Reference

Part V provides reference information for Real Application Clusters.

### Appendix A, "Troubleshooting"

This appendix describes how to use trace files for troubleshooting Oracle installation issues. It also explains how to contact Oracle Support Services.

### Appendix B, "Associating Instances and Users with Free Lists and Free List Groups (Optional)"

This appendix explains how to associate instances and users with free lists and free list groups. It also discusses SQL-specific free list options and the preallocation of extents to free list groups.

### Appendix C, "Real Application Clusters Management Tools Error Messages"

This appendix explains describes the error messages for the Real Application Clusters Management tools.

### Glossary

The glossary defines terms used in this book as well as terms relevant to the subject matter of this book.

## Related Documentation

For more information, see these Oracle resources:

- *Oracle9i Real Application Clusters Documentation Online Roadmap*
- *Oracle9i Real Application Clusters Concepts*
- *Oracle9i Real Application Clusters Setup and Configuration*
- *Oracle9i Real Application Clusters Deployment and Performance*
- *Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration*
- *Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration* on the Real Application Clusters Guard II software CD
- Your platform-specific Oracle Real Application Clusters Guard installation guide

### Installation Guides

- *Oracle9i Installation Guide Release 2 (9.2.0.1) for UNIX Systems: AIX-Based Systems, Compaq Tru64, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris*
- *Oracle9i Database Installation Guide for Windows*
- *Oracle Diagnostics Pack Installation*
- Oracle Real Application Clusters Guard I installation manuals; there are several titles available and they are platform-specific
- *Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration* on the Real Application Clusters Guard II software CD

### Operating System-Specific Administrative Guides

- *Oracle9i Administrator's Reference Release 2 (9.2.0.1) for UNIX Systems: AIX-Based Systems, Compaq Tru64, HP 9000 Series HP-UX, Linux Intel, and Sun Solaris*
- *Oracle9i Database Administrator's Guide for Windows*
- *Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration*
- *Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration* on the Real Application Clusters Guard II software CD

### Oracle9*i* Real Application Clusters Management

- *Oracle Enterprise Manager Administrator's Guide*
- *Getting Started with the Oracle Diagnostics Pack*

## Generic Documentation

- *Oracle9i Database Concepts*
- *Oracle9i Net Services Administrator's Guide*
- *Oracle9i Database New Features*
- *Oracle9i Database Reference*

Many of the examples in this book use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle9i Sample Schemas* for information about how these schemas were created and how to use them.

In North America, printed documentation is available for sale in the Oracle Store at `http://oraclestore.oracle.com/`

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

`http://www.oraclebookshop.com/`

Other customers can contact their Oracle representative to purchase printed documentation.

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/admin/account/membership.html`

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/docs/index.htm`

To access the database documentation search engine directly, please visit

`http://tahiti.oracle.com`

# 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**. |

| Convention | Meaning | Example |
|---|---|---|
| *Italics* | Italic typeface indicates book titles or emphasis. | *Oracle9i Database Concepts*<br>Ensure that the recovery catalog and target database do *not* 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.<br>You can back up the database by using the `BACKUP` command.<br>Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view.<br>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.<br>**Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown. | Enter `sqlplus` to open SQL*Plus.<br>The password is specified in the `orapwd` file.<br>Back up the datafiles and control files in the `/disk1/oracle/dbs` directory.<br>The `department_id`, `department_name`, and `location_id` columns are in the `hr.departments` table.<br>Set the `QUERY_REWRITE_ENABLED` initialization parameter to `true.`<br>Connect as `oe` user.<br>The `JRepUtil` class implements these methods. |
| *`lowercase italic monospace (fixed-width) font`* | Lowercase italic monospace font represents placeholders or variables. | You can specify the *`parallel_clause`*.<br>Run `U`*`old_release`*`.SQL` where *`old_release`* 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 (*digits* [ , *precision* ]) |
| { } | 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}<br>[COMPRESS \| NOCOMPRESS] |
| . . . | Horizontal ellipsis points indicate either: | |
| | ■ That we have omitted parts of the code that are not directly related to the example | CREATE TABLE ... AS *subquery*; |
| | ■ That you can repeat a portion of the code | SELECT *col1*, *col2*, ... , *coln* FROM employees; |
| .<br>.<br>. | Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example. | SQL> SELECT NAME FROM V$DATAFILE;<br>NAME<br>-----------------------------------<br>/fsl/dbs/tbs_01.dbf<br>/fs1/dbs/tbs_02.dbf<br>.<br>.<br>.<br>/fsl/dbs/tbs_09.dbf<br>9 rows selected. |
| Other notation | You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown. | acctbal NUMBER(11,2);<br>acct    CONSTANT NUMBER(4) := 3; |
| *Italics* | Italicized text indicates placeholders or variables for which you must supply particular values. | CONNECT SYSTEM/*system_password*<br>DB_NAME = *database_name* |

| Convention | Meaning | Example |
|---|---|---|
| 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;<br>SELECT * FROM USER_TABLES;<br>DROP TABLE hr.employees; |
| lowercase | Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.<br>**Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown. | SELECT last_name, employee_id FROM employees;<br>sqlplus hr/hr<br>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 Databae Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > 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 |

| Convention | Meaning | Example |
|---|---|---|
| `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 *command prompt* 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\"`<br>`C:\>imp SYSTEM/`*`password`*` FROMUSER=scott TABLES=(emp, dept)` |
| *`HOME_NAME`* | 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`*`HOME_NAME`*`TNSListener` |

| Convention | Meaning | Example |
|---|---|---|
| *ORACLE_HOME* and *ORACLE_ BASE* | In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory that by default used one of the following names:<br>■ C:\orant for Windows NT<br>■ C:\orawin95 for Windows 95<br>■ C:\orawin98 for Windows 98<br>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* 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 nn is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*.<br>All directory path examples in this guide follow OFA conventions.<br>Refer to *Oracle9i Database Getting Started for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories. | Go to the *ORACLE_BASE*\*ORACLE_ HOME*\rdbms\admin directory. |

# 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 Corporation 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 Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

# What's New in Real Application Clusters Administration?

This section describes the new administrative features of Oracle9*i* release 2 (9.2) for **Real Application Clusters**.

> **See Also:** *Oracle9i Real Application Clusters Concepts* for a detailed explanation of this product's new features

The topic in this section is:

- Oracle9i Release 2 New Features for the Administration of Real Application Clusters Databases

# Oracle9*i* Release 2 New Features for the Administration of Real Application Clusters Databases

- **Recovery Manager Enhancements for Real Application Clusters**

  Autolocation for Real Application Clusters: RMAN automatically discovers which nodes of an Oracle Real Application Clusters configuration can access the files that you want to back up or restore. RMAN autolocates the following files:

  - Backup pieces and datafile or control file copies during backup or restore
  - Archived redo logs during backup

  RMAN enables the autolocation feature whenever the allocated channels have different PARMS or CONNECT strings.

- **Real Application Clusters Guard II**

  Real Application Clusters Guard II supports comprehensive workload management to maintain high availability for the Real Application Clusters database and its applications. You can use the standard Server Control (SRVCTL) Utility and the Database Configuration Assistant (DBCA) to administer Real Application Clusters Guard II.

  > **See Also:** *Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration* on the Real Application Clusters Guard II software CD for more information about administering Real Application Clusters Guard II with SRVCTL and the DBCA

- **Real Application Clusters Management Tools Error Messages**

  The error messages for the Real Application Clusters management tools are documented in Appendix C, "Real Application Clusters Management Tools Error Messages". The message groups in this appendix include the following:

  - Global Services Daemon Error Messages (PRKA)
  - Cluster Command Error Messages (PRKC)
  - Cluster Setup Errors for Windows NT (PRKI)
  - Server Control (SRVCTL) Utility Error Messages (PRKO)
  - Cluster Database Management Messages (PRKP)
  - Repository Errors (PRKR)

# Part I

# Introduction to Administering Real Application Clusters

Part I provides information on parameter management, storage component administration, and general administrative procedures in Real Application Clusters. The chapters in Part I are:

- Chapter 1, "Introduction to Real Application Clusters Administration"
- Chapter 2, "Parameter Management in Real Application Clusters Environments"
- Chapter 3, "Administering Storage Components in Real Application Clusters"
- Chapter 4, "Administering Real Application Clusters Databases with the Server Control Utility, SQL, and SQL*Plus"

# 1

# Introduction to Real Application Clusters Administration

This chapter provides an overview of the processes and components involved in administering **Real Application Clusters**. This chapter includes the following topics:

- Administering Real Application Clusters Databases
- Parameter Management in Real Application Clusters
- Administering Storage in Real Application Clusters
- General Administration Issues
- Oracle Enterprise Manager in Real Application Clusters
- Backup and Recovery in Real Application Clusters
- Real Application Clusters Scalability
- Troubleshooting

# Administering Real Application Clusters Databases

This book describes Real Application Clusters-specific administrative duties. These tasks are in addition to those described in the *Oracle9i Database Administrator's Guide* which you should read before you use this book to administer Real Application Clusters. The following is an overview of the Real Application Clusters-specific administrative issues.

# Parameter Management in Real Application Clusters

You can establish **cluster database**-wide parameter settings and specify instance-specific settings using the **server parameter file**. or you can use client-side parameter files to manage parameter settings.

There are also three types of parameters in Real Application Clusters. Some parameters can have different values for one or more instances while others must be either the same for all instances or unique among all instances. Parameter settings also have unique effects on **instance** startup within Real Application Clusters databases.

**See Also:** Chapter 2, "Parameter Management in Real Application Clusters Environments"

# Administering Storage in Real Application Clusters

There are Real Application Clusters-specific storage management issues to consider such as datafile administration and undo tablespace management. You must also understand redo log file management and the use of threads of redo in Real Application Clusters instances. Automatic undo management simplifies the administration of storage management for undo tablespaces.

**See Also:** Chapter 3, "Administering Storage Components in Real Application Clusters"

# General Administration Issues

You can start and stop Real Application Clusters instances and databases using SQL, SQL*Plus, and command-line utilities. You can also set and connect to remote instances and perform administrative operations by way of Oracle Net. However, SQL and SQL*Plus commands can have both *local* and *global* effects on instances.

You can also use the SRVCTL utility to perform many administrative duties in Real Application Clusters databases. You can also use SRVCTL to perform other tasks such as managing configuration information, and to delete, rename, and move instances.

> **See Also:** Chapter 4, "Administering Real Application Clusters Databases with the Server Control Utility, SQL, and SQL*Plus"

## Oracle Enterprise Manager in Real Application Clusters

You can use **Oracle Enterprise Manager** to perform most administrative duties in Real Application Clusters. Use Oracle Enterprise Manager to start and stop instances, monitor their performance, schedule jobs, and to perform backup and recovery operations.

> **See Also:** Chapter 5, "Administering Real Application Clusters Databases with Oracle Enterprise Manager"

## Backup and Recovery in Real Application Clusters

You can configure **Recovery Manager (RMAN)** for use in Real Application Clusters. You can also configure RMAN to automatically back up each control file and to automatically save channel settings.

You can use RMAN or operating system commands to perform both open and closed backups in Real Application Clusters. Your recovery options in Real Application Clusters include complete, incomplete, and block media recovery. You can also use parallel recovery.

> **See Also:**
>
> - Chapter 6, "Using Recovery Manager in Real Application Clusters Environments"
> - Chapter 7, "Performing Backup and Recovery in Real Application Clusters Environments"

## Real Application Clusters Scalability

To meet increased system demands or replace failed hardware, you can add **nodes** and instances and delete instances in Real Application Clusters. Use the **Oracle Universal Installer (OUI)** and the **Database Configuration Assistant (DBCA)** to dynamically add nodes and to add and remove instances.

**See Also:** Chapter 8, "Adding Nodes and Instances and Deleting Instances in Real Application Clusters"

## Troubleshooting

Oracle records processing information in various trace and log files. You can refer to these files for troubleshooting information and you should regularly copy these files to avoid overwriting them.

**See Also:** Chapter A, "Troubleshooting" for more information about these files

# 2

# Parameter Management in Real Application Clusters Environments

This chapter describes how to administer the server parameter and client-side parameter files. It also describes parameter use in **Real Application Clusters** and how parameters affect startup processing. The topics in this chapter are:

> **See Also:** *Oracle9i Real Application Clusters Setup and Configuration* for information on creating and configuring the server parameter file in Real Application Clusters

# Administering the Server Parameter File in Real Application Clusters Databases

Oracle uses parameter settings in the **server parameter file** to control database resources. You can also use the traditional client-side parameter files, however, Oracle Corporation recommends that you use the server parameter file. This section describes administering the server parameter file and includes the following topics:

- Backing Up the Server Parameter File
- Setting the Server Parameter File Parameter Values for Real Application Clusters
- Exporting the Server Parameter File For Backward Compatibility

## Backing Up the Server Parameter File

Oracle Corporation recommends that you regularly create copies of the server parameter file for recovery purposes. Do this using the `CREATE PFILE FROM SPFILE` statement.

You can also recover your database's server parameter file by starting up an **instance** using a client-side **initialization parameter file**. Then re-create the server parameter file using the `CREATE SPFILE` statement.

> **See Also:** *Oracle9i SQL Reference* for more information about the `CREATE PFILE` and `CREATE SPFILE` statements

## Setting the Server Parameter File Parameter Values for Real Application Clusters

Oracle automatically updates the values in the server parameter file for parameter settings that you change using **Oracle Enterprise Manager** or `ALTER SYSTEM SET` statements. In addition, the `ALTER SYSTEM RESET` syntax enables you to *undo* the effects of parameter settings in the server parameter file and parameters that you manually set.

For example, assume you start an instance with a server parameter file containing the following entries:

```
*.OPEN_CURSORS=500
proddb1.OPEN_CURSORS=1000
```

**Note:** Sample settings only appear as text in these examples. However, the server parameter file is a binary file.

For the instance with *sid* *proddb1*, the parameter remains set to 1000 even though it is preceded by a database-wide setting of 500. The instance-specific parameter setting in the parameter file prevents database-wide alterations of the setting. This gives the Database Administrator (DBA)of instance *proddb1* complete control over parameter settings for that instance. These two types of settings can appear in any order in the parameter file.

If another DBA executes the following:

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

Then Oracle updates the setting on all instances except the instance with *sid* *proddb1*. If you later change the parameter setting by executing the following on the instance with *sid* *proddb1*, then the parameter begins accepting future ALTER SYSTEM values set by other instances:

```
ALTER SYSTEM RESET OPEN_CURSORS SCOPE=MEMORY sid='proddb1';
```

Then execute the following on another instance and the instance with *sid* *proddb1* also assumes the new setting of 2000:

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

If the server parameter file contains the entries:

```
proddb1.OPEN_CURSORS=1000
*.OPEN_CURSORS=500
```

Executing:

```
ALTER SYSTEM RESET OPEN_CURSORS SCOPE=SPFILE sid='proddb1';
```

Makes Oracle disregard the first entry from the server parameter file.

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

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

**Note:** Not all ALTER SYSTEM statement changes and modifications are recorded in the server parameter file. Some ALTER SYSTEM statements only change *in-memory* parameter settings.

## Exporting the Server Parameter File For Backward Compatibility

You can revert to a pre-Real Application Clusters release of Oracle cluster software and convert from using the server parameter file to the traditional client-side parameter file type. Do this by using the FROM option of the CREATE PFILE statement. Export the contents of a server parameter file into a prerelease 1 (9.0.1) release parameter file using the following syntax:

```
CREATE PFILE[= 'pfile-name'] FROM SPFILE [='raw_device_name'];
```

This statement exports the contents of the server parameter file into a prerelease 1 (9.0.1) release initialization parameter file named 'pfile-name'. If you do not specify a PFILE or an SPFILE file name, Oracle uses the platform-specific default PFILE and SPFILE names.

Oracle creates the PFILE as a text file on the server. This file contains all parameter settings of all instances. Entries for overrides appear as *sid*.parameter=value. The PFILE also contains any comments associated with the parameter. Comments appear in the same line as the parameter setting. You must move any *sid*-specific entries to an instance-specific parameter file and delete the *sid* specification. The CREATE PFILE statement requires DBA privileges.

You can execute the CREATE PFILE statement to:

- Create backups of the server parameter file. Note that **Recovery Manager (RMAN)** *does not* backup the server parameter file.
- Obtain a list of all parameter values currently used by an instance for diagnostic purposes. Do this using the SHOW PARAMETER command or by querying the V$PARAMETER table.
- Modify the server parameter file by exporting it, editing the output file, and then re-creating it.

## Setting Parameters within the Server Parameter File

Use the *sid* designator to set instance-specific parameter values in the server parameter file. For example, the following:

```
proddb1.OPEN_CURSORS = 1000
proddb2.OPEN_CURSORS = 1500
```

Sets `OPEN_CURSORS` to `1000` for instance *`proddb1`*, and to `1500` for instance *`proddb2`*. These entries are recognized as entries for specific *sid*`s` in a Real Application Clusters database. The value `1000` is applied to the parameter when the instance is started up with *sid* *`proddb1`* and the value `1500` is applied to the parameter when the instance is started up with *sid* *`proddb2`*.

The parameter file entry:

```
*.DB_FILE_MULTIBLOCK_READ_COUNT=16
```

Sets the value of parameter `DB_FILE_MULTIBLOCK_READ_COUNT` to `16` for all instances. Parameter `DB_FILE_MULTIBLOCK_READ_COUNT` takes the value `16` for all instances because a *sid* of `*` is considered global and the value specified in the parameter setting is applied to all the *sid*`s`.

Note that the server parameter file supports the prerelease 1 (9.0.1) syntax of:

```
parameter1 = value1
```

In addition, `parameter1` takes the value `value1` regardless of the *sid*. You can override parameters for specific *sid*`s` as follows:

```
OPEN_CURSORS = 1000
proddb1.OPEN_CURSORS = 1500
```

In this case, `OPEN_CURSORS` takes the value `1000` for all the instances that have a *sid* other than *`proddb1`* and takes the value 1500 in the instance with *sid* *`proddb1`*.

### Specifying Comments in the Server Parameter File

Specify comments with parameter settings on the same line with the parameter setting. For example, if `init.ora` contains the following lines:

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

The string `second comment` is associated with `OPEN_CURSORS`'s setting. 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.

**Note:** Use caution when modifying or adding values to the server parameter file. Doing this incorrectly can corrupt the file and prevent database start-up. To repair the file, you may need to create the PFILE and regenerate the SPFILE.

# Using Client-Side Parameter Files

You can use one or more client-side parameter files to manage parameter settings in Real Application Clusters. By default, if you do not specify `PFILE` in your `STARTUP` command, Oracle uses a server parameter file.

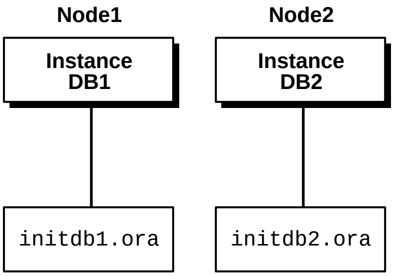
You can set global parameters within instance-specific parameter files. To do this, you must have identical parameter settings for global parameters in all of your instance-specific parameter files. You can also maintain one file for global parameters and point to it with the **IFILE** parameter.

**Note:** If you use client-side parameter files, self-tuning parameter changes that Oracle makes are lost after shutdown. In addition, using client-side parameter files increases your parameter administration overhead.

## Client-Side Parameter File Naming Conventions

Base your file names for the client-side parameter filenames on the *sid* of each instance and the **global database name**. For example, name each instance-specific parameter file **initsid.ora**, where *sid* is the system identifier of the instance. Name the common parameter file, **initdbname.ora**, where `db_name` is the database name of your Real Application Clusters database as shown in Figure 2–1.

***Figure 2–1 Traditional Instance-Specific Initialization Files***



The parameter file can contain both instance-specific and global parameter settings. If you include global parameter settings, the entries for these must be identical in each instance's file.

Oracle reads entries in this file beginning at the top of the file. When interpreting parameters, Oracle uses the last value for any parameters in this file that are duplicates. This is true for both instance-specific and global parameter settings. Table 2–1 describes the optional initialization parameter files:

***Table 2–1 Naming Conventions and Descriptions for Optional Parameter Files***

| **Initialization Parameter File/ Naming Convention** | **Description** |
|---|---|
| Instance-specific file:<br>`init`*`sid`*`.ora` | Each node's instance can have its own `init`*`sid`*`.ora` file, where `sid` is the **Oracle system identifier (sid)** of the instance. This file uniquely defines the instance-specific parameter settings. Use the `IFILE` parameter from within this file to call the `init`*`db_name`*`.ora` file. |
| Global file:<br>`init`*`db_name`*`.ora` | If you do not include global parameter file settings in each instance-specific file, you must store common parameters in an `init`*`db_name`*`.ora` file, where *`db_name`* is the database name. It lists the common database parameters shared by all instances. A copy of this file must reside on each instance in the database cluster. If you use a cluster file system (CFS), all instances that run from the same Oracle home share this file. |

## Purpose of init*sid*.ora

The `init`*`sid`*`.ora` file uses the `IFILE` parameter to point to the common file for common parameter settings. The `init`*`sid`*`.ora` file defines the following for each instance:

- Unique instance name
- Unique thread number and **instance number**
- Private rollback segments or undo tablespaces
- Local **listener** if not using default TCP/IP port address for the listener

The convention for deriving a `sid` is to use the value of the `DB_NAME` parameter in the `init`*`db_name`*`.ora` file and the **thread number**. For example, if the `DB_NAME` is `db`, and the first instance has a `thread ID` of `1`, its *sid* is `db1`; the second instance uses the *sid* `db2` to identify its instance; and so on. This is the logic that the DBCA uses when it derives a sid. A sid, however, can have any value you choose.

Example 2–1 and Example 2–2 show the contents of `init`*`sid`*`.ora` files for two instances for each **node** numbered `1` and `2` respectively:

***Example 2–1 initdb1.ora***

```
ifile='C:\OracleSW\admin\db\pfile\initdb.ora'
thread=1
instance_name=db1
instance_number=1
```

***Example 2–2 initdb2.ora***

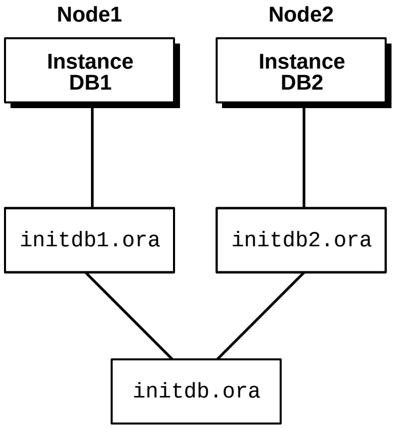
```
ifile='C:\OracleSW\admin\db\pfile\initdb.ora'
thread=2
instance_name=db2
instance_number=2
```

> **See Also:** *Oracle9i Database Reference* for complete parameter descriptions

## Purpose of init*db_name*.ora

Name the `init`*`db_name`*`.ora` parameter file using the `IFILE` parameter setting in each `init`*`sid`*`.ora` file as shown in Figure 2–2.

***Figure 2–2 Common Initialization File Example***



All instances must use the same common file if you use the traditional parameter files.

Example 2–3 shows an `init`*`db_name`*`.ora` file (`initdb.ora`) created for a **hybrid** or General Purpose database:

***Example 2–3 initdb.ora***

```
db_name="db"
db_domain=us.acme.com
cluster_database=true
service_names=db.us.acme.com
db_files=1024  # INITIAL
control_files=("\\.\db_control1", "\\.\db_control2")
open_cursors=100
db_file_multiblock_read_count=8  # INITIAL
db_block_buffers=13816  # INITIAL
shared_pool_size=19125248  # INITIAL
large_pool_size=18087936
java_pool_size=2097152
```

```
log_checkpoint_interval=10000
log_checkpoint_timeout=1800
processes=50  # INITIAL
parallel_max_servers=5  # SMALL
log_buffer=32768  # INITIAL
max_dump_file_size=10240  # limit trace file size to 5M each
global_names=true
oracle_trace_collection_name=""
background_dump_dest=C:\OracleSW\admin\db\bdump
user_dump_dest=C:\OracleSW\admin\db\udump
db_block_size=4096
remote_login_passwordfile=exclusive
os_authent_prefix=""
dispatchers="(protocol=TCP)(lis=listeners_db)"
compatible=9.2
sort_area_size=65536
sort_area_retained_size=65536
```

**Note:** `DB_BLOCK_BUFFERS` will be obsolete in a future Oracle release. If you set `DB_BLOCK_BUFFERS`, Oracle uses the value but also records a warning in your alert log file.

**See Also:** *Oracle9i Database Reference* for complete parameter descriptions

## Placement and Use of IFILE Parameters in Instance-Specific Files

If you duplicate parameter entries in a parameter file, the last value specified in the file for the parameter overrides previous values. To ensure Oracle uses the correct common parameter values, place the `IFILE` parameter at the end of any instance-specific parameter files. Conversely, you can override common parameter values by placing the `IFILE` parameter before the instance-specific parameter setting.

**Note:** The DBCA places the `IFILE` parameter at the top of the parameter file.

## Using Multiple IFILE Entries in Client-Side Parameter Files

You can specify IFILE more than once in your initial parameter file to include multiple global parameter files. However, do not accidentally reset a parameter value in subsequent common parameter files. Otherwise, each subsequent entry in the files specified by IFILE overrides previous values. For example, an instance-specific parameter file might include an init_dbname.ora file and separate parameter files for other parameter settings as in this example:

```
IFILE=INIT_CLUSTER.ORA
IFILE=INIT_LOG.ORA
IFILE=INIT_GC.ORA
LOG_ARCHIVE_START=FALSE
THREAD=3
UNDO_MANAGEMENT=AUTO
```

In this example, the value of LOG_ARCHIVE_START=false overrides any value specified in the parameter file INIT_LOG.ORA for this parameter. This is because the LOG_ARCHIVE_START parameter appears after the IFILE entry.

**See Also:**

- "Parameters That Must Be Identical Across All Instances" on page 2-13
- Chapter 4, "Administering Real Application Clusters Databases with the Server Control Utility, SQL, and SQL*Plus" for more information on starting and stopping instances

# Parameter Settings in Real Application Clusters

As mentioned, some parameters must be identical across all instances in Real Application Clusters. Other parameters can have unique values within each instance.

## Unique Identification of Instances with Parameters in the Sever Parameter File

Each instance has several elements or components whose characteristics are controlled by parameter settings. Some of these are database objects and others are resource-like components that facilitate inter-instance processing. You uniquely identify these instance components using the sid designator in the server parameter file or by using parameter settings in `init`*`sid`*`.ora.`

> **See Also:** *Oracle9i Database Reference* for more information about initialization parameters

For example, Table 2–2 shows the `sids and instance names` if the database name is `db` and the thread IDs for each instance are `1`, `2`, and `3` respectively:

***Table 2–2 Example sids and Instance Names***

| thread id | sid | instance_name |
|---|---|---|
| 1 | db1 | db1 |
| 2 | db2 | db2 |
| 3 | db3 | db3 |

## Instance-Specific Parameter Settings Using sid in the Server Parameter File

In the server parameter file, use the *sid* designator to identify instance-specific settings. Also use the *sid* designator in the server parameter file when you create instances that specify:

- `INSTANCE_NAME`
- `THREAD`

# Types of Parameters in Real Application Clusters

There are three types of initialization parameters in Real Application Clusters environments as described in this section. There are:

- Multi-Valued Parameters
- Parameters That Must Be Identical Across All Instances
- Parameters That Must Be Unique Across All Instances

> **See Also:** *Oracle9i Database Reference* for details about other Oracle initialization parameters

## Multi-Valued Parameters

You can configure some parameters to have different values for one or more instances. Parameters that can be multi-valued have a default value and a value for each instance that has modified the default setting.

Use the `ALTER SYSTEM SET` statement to set multiple values for such parameters. You can also use `ALTER SYSTEM SET` to define a global value that is effective for all instances. In addition, you can override these global values for specific instances.

## Parameters That Must Be Identical Across 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 Real Application Clusters. Specify these parameter values in the common parameter file, or within each `init_dbname.ora` file on each instance. The following list shows the parameters that must be identical on every instance.

- `CONTROL_FILES`
- `DB_BLOCK_SIZE`
- `DB_FILES`
- `DB_NAME`
- `DB_DOMAIN`
- `ARCHIVE_LOG_TARGET`
- `ROW_LOCKING`
- `DML_LOCKS` (Only if set to zero)
- `LOG_ARCHIVE_DEST_n` (Optional)
- `MAX_COMMIT_PROPAGATION_DELAY`
- `SERVICE_NAMES`

- ACTIVE_INSTANCE_COUNT
- TRACE_ENABLED
- GC_FILES_TO_LOCKS (Optional: You do not need to set this parameter because Oracle automatically controls resource assignments.)

> **See Also:** *Oracle9i Real Application Clusters Deployment and Performance* for more information about setting GC_FILES_TO_LOCKS

## Parameters That Must Be Unique Across All Instances

If you use the THREAD or ROLLBACK_SEGMENTS parameters, Oracle Corporation recommends setting unique values for them by using the sid identifier in the server parameter file. 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 also uses INSTANCE_NUMBER to assign free space to instances using the INSTANCE option of the ALLOCATE EXTENT clause in the ALTER TABLE or ALTER CLUSTER statements.
- Specify the THREAD parameter so instances avoid the overhead of acquiring different thread numbers during startup and shutdown. 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.
- Specify INSTANCE_NAME to uniquely identify the instance. The default is the instance's sid and Oracle Corporation recommends that you use this for INSTANCE_NAME.
- Oracle acquires private rollback segments upon instance startup based on the rollback segment names you identify with the ROLLBACK_SEGMENTS initialization parameter. If you do not declare rollback segment names with this parameter for an instance, Oracle acquires public rollback segments for the instance.
- If you specify UNDO_TABLESPACE with automatic undo management enabled, set this parameter to a unique value for each instance.

# Considerations for Parameters in Clusters

Table 2–3 summarizes the considerations for using certain parameters in Real Application Clusters databases. Table 2–3 lists the parameters in alphabetical order.

*Table 2–3 Initialization Parameter Notes for Instances in Clusters*

| Parameter | Description and Comments |
|---|---|
| CLUSTER_ DATABASE | To enable a database to be started in Real Application Clusters mode, set this parameter to TRUE. |
| CLUSTER_ DATABASE_ INSTANCES | Set this parameter to the number of instances in your Real Application Clusters environment. A proper setting for this parameter can improve memory use. |

***Table 2–3 (Cont.) Initialization Parameter Notes for Instances in Clusters***

| Parameter | Description and Comments |
|---|---|
| CLUSTER_ INTERCONNECTS<br>See Also: Your platform-specific documentation for more information. | The CLUSTER_INTERCONNECTS parameter is supported on some platforms. Refer to your platform-specific documentation for the specific use of this parameter, its syntax, and its behavior. In general, however, you should not have to set CLUSTER_INTERCONNECTS.<br>You do not need to set this parameter if you have a single cluster interconnect. You also do not need to set it if the default cluster interconnect meets the bandwidth requirements of your Real Application Clusters database(s), which is typically the case.<br>Oracle uses information from CLUSTER_INTERCONNECTS to distribute interconnect traffic among the various network interfaces if you specify more than 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.<br>In rare cases where a single cluster interconnect cannot meet your bandwidth requirements, consider setting CLUSTER_INTERCONNECTS. For example, you might need to set this parameter in some Data Warehouse environments with very high interconnect bandwidth demands from one or more database(s).<br>For example, if you have two databases with high interconnect bandwidth requirements, then you can override the default interconnect(s) provided by your operating system and nominate a different interconnect for each database using the following syntax in each server parameter file:<br>Database One: CLUSTER_INTERCONNECTS = ip1<br>Database Two: CLUSTER_INTERCONNECTS = ip2<br>Where ip*n* is an IP address in standard dotted-decimal format, for example, 144.25.16.214.<br>However, if you have one database with very high bandwidth demands, then you can nominate multiple interconnects, for example, using the following syntax:<br>CLUSTER_INTERCONNECTS = ip1:ip2:...:ipn<br>If you set multiple values for CLUSTER_INTERCONNECTS as in the previous 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.<br>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. |
| DB_NAME | If you set a value for DB_NAME in instance-specific parameter files, it must be identical for all instances. |

*Table 2–3 (Cont.) Initialization Parameter Notes for Instances in Clusters*

| Parameter | Description and Comments |
|---|---|
| DISPATCHER | To enable a **shared server** configuration, set the DISPATCHERS parameter. The DISPATCHERS parameter may contain many attributes.<br>Oracle Corporation recommends that you configure at least the PROTOCOL and LISTENER attributes. PROTOCOL specifies the network protocol for which the **dispatcher** generates a listening end point. LISTENER specifies an alias name for the listeners with which the PMON process registers dispatcher information. Set the alias to a name that is resolved through a naming method such as a **tnsnames.ora file**.<br>*Oracle9i Net Services Administrator's Guide* for complete information about configuring the DISPATCHER parameter and its attributes and for configuring the shared server |
| DML_LOCKS | Must be identical on all instances only if set to zero. The default value assumes an average of four tables referenced per transaction. For some systems, this value may not be enough. If you set the value of DML_LOCKS to 0, **enqueues** are disabled and performance is slightly increased. However, you cannot use DROP TABLE, CREATE INDEX, or explicit lock statements such as LOCK TABLE IN EXCLUSIVE MODE. |
| INSTANCE_NAME | If specified, this parameter must have unique values on all instances. In Real Application Clusters environments, all instances can be associated with a single database service. Clients can override **connection load balancing** by specifying a particular instance by which to connect to the database. INSTANCE_NAME specifies the unique name of this instance. Oracle Corporation recommends that you set INSTANCE_NAME equivalent to the sid. |
| LOG_ARCHIVE_FORMAT | This parameter is applicable only if you are using the redo log in ARCHIVELOG mode. Use a text string and variables to specify the default filename format when archiving redo log files. The string generated from this format is appended to the string specified in the LOG_ARCHIVE_DEST_*n* parameter. You must include the thread number.<br>The following variables can be used in the format:<br>▪ %s: log sequence number<br>▪ %S: log sequence number, zero filled<br>▪ %t: thread number<br>▪ %T: thread number, zero filled<br>Using uppercase letters for the variables (for example, %S) causes the value to be fixed length and padded to the left with zeros. An example of specifying the archive redo log filename format is:<br>▪ LOG_ARCHIVE_FORMAT = "LOG%s_%t.ARC" |

*Table 2–3 (Cont.) Initialization Parameter Notes for Instances in Clusters*

| Parameter | Description and Comments |
|---|---|
| MAX_COMMIT_PROPAGATION_DELAY | This is a Real Application Clusters-specific parameter. However, you should not change it except under a limited set of circumstances.<br>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 lock value when getting the snapshot SCN for a query. Units are in hundredths of seconds. Under unusual circumstances involving rapid updates and queries of the same data from different instances, the SCN might not be refreshed in a timely manner. Setting the parameter to zero causes the SCN to be refreshed immediately after a commit. The default value (700 hundredths of a second, or seven seconds) is an upper bound that enables the preferred existing high performance mechanism to remain in place.<br>If you want commits to be seen immediately on remote instances, you may need to change the value of this parameter. |
| NLS_* parameters<br>For Oracle Globalization Support | There are several **Globalization Support** parameters as described in *Oracle9i Database Reference* and *Oracle9i Database Globalization Support Guide*. You can set different values for different instances. |
| PROCESSES | Defaults for the SESSIONS and TRANSACTIONS parameters are derived directly or indirectly from the value of the PROCESSES parameter. Therefore, if you change the value of PROCESSES, you should evaluate whether to adjust the values of those derived parameters. If you do not use defaults, you may want to increase the values for some of the previous parameters in this table to allow for additional background processes. |
| RECOVERY_PARALLELISM | To speed up roll forward or cache recovery processing, you may want to set this parameter to specify the number of processes to participate in instance or failure recovery. A value of zero or one indicates that recovery is to be performed serially by one process. |
| ROLLBACK_SEGMENTS<br>(Use only in Rollback Managed Undo Mode)<br>**Note:** Oracle Corporation strongly recommends that you use automatic undo management, not Rollback Managed Undo. | Use this parameter in manual rollback managed undo mode only to specify the private rollback segments for each instance by allocating one or more rollback segments by name to an instance. If you set this parameter, the instance acquires all of the rollback segments named in this parameter, even if the number of rollback segments exceeds the minimum number required by the instance, calculated from the ratio of:<br>TRANSACTIONS / TRANSACTIONS_PER_ROLLBACK_SEGMENT |

*Table 2–3 (Cont.) Initialization Parameter Notes for Instances in Clusters*

| Parameter | Description and Comments |
|---|---|
| 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. |
| SPFILE | The value of SPFILE is the name of the current server parameter file in use. You can define the SPFILE parameter in a client-side PFILE to indicate the name of the server parameter file to use.<br>When the server uses the default server parameter file, the server internally sets the value of SPFILE. |
| THREAD | If specified, this parameter must have unique values on all instances. THREAD is a Real Application Clusters parameter that specifies the number of the redo thread to be used by an instance.<br>In Real Application Clusters, you can specify any available redo thread number as long as that thread number is enabled and is not used by another instance. Although not recommended, using a value of zero specifies that an instance can use any available, enabled public thread. |

**See Also:** *Oracle9i Database Reference* for more information about these parameters and *Oracle9i Real Application Clusters Deployment and Performance* for a discussion of additional parameters for parallel execution in Real Application Clusters environments

## The Startup Process and Parameters in Real Application Clusters

In Real Application Clusters, the first instance to start mounts the database. In addition, entries in the alert.log file of the first instance to start identifies that instance as the first one to start.

**Note:** To find your alert log file, use the search string alert*.log. On Windows platforms, the SID replaces the asterisk in the format *alert.log. You can usually find alert.log in the background_dump_dest directory.

**See Also:** Chapter 4, "Administering Real Application Clusters Databases with the Server Control Utility, SQL, and SQL*Plus" for more information about starting instances

## Special Startup Considerations for Traditional Parameter File Use

If you use the traditional parameter files and a file for an instance contains a global parameter, its value must match the value set in other instances for that parameter. Otherwise, the instance cannot mount the database.

## Starting Two Instances on Remote Nodes

Oracle Corporation recommends using the **Server Control (SRVCTL) Utility** utility to start instances. You can also use SRVCTL for other administrative tasks as described under the heading "Administering Real Application Clusters Environments with SRVCTL" on page 4-2. The rest of this section describes using SQL*Plus to start instances.

To start multiple instances from a SQL*Plus session on one node by way of Oracle Net. For example, you can use a SQL*Plus session on a local node to start two instances on remote nodes using individual parameter files named `init_db1.ora` and `init_db2.ora`. Before connecting to the database, in SQL*Plus direct your commands to the first instance by entering:

```
SET INSTANCE DB1;
```

Connect to the first instance, start it, and disconnect from it by entering:

```
CONNECT / AS SYSDBA;
STARTUP PFILE=$ORACLE_HOME/dbs/initsid.ora
DISCONNECT;
```

Where the file init*sid*.ora contains an entry for an `spfle.ora` file's location on a raw device.

redirect commands to the second instance using the following syntax:

```
SET INSTANCE DB2;
```

Connect to and start the second instance by entering:

```
CONNECT / AS SYSDBA;
STARTUP PFILE=full pathINIT_DB2.ORA;
```

Here, `DB1` and `DB2` are *sid*s for the two instances. These *sid*s are defined with the *sid* entry in `tnsnames.ora`. An SPFILE parameter entry in the init*sid*.ora file specifies a location on the remote instance.

### IFILE Use in Traditional Parameter File Scenarios

In the previous example, both parameter files can use the IFILE parameter to include values from an init_dbname.ora file.

## Setting Instance Numbers

You must explicitly specify an instance number by using the INSTANCE_NUMBER parameter  upon startup. You can do this with Real Application Clusters enabled or disabled. Oracle Corporation recommends that you set INSTANCE_NUMBER equal to the value you set to identify the instance's THREAD. The SQL*Plus command:

```
SHOW PARAMETER INSTANCE_NUMBER
```

Shows the current number for each instance.

**Note:** All instance numbers must be unique.

If you start an instance merely to perform administrative operations with Real Application Clusters disabled, you can omit the INSTANCE_NUMBER parameter from the parameter file. An instance starting with Real Application Clusters disabled can also specify a thread other than 1 to use the online redo log files associated with that thread.

**See Also:**

- Chapter 3, "Administering Storage Components in Real Application Clusters" for more information about managing undo space in Real Application Clusters
- *Oracle9i Real Application Clusters Deployment and Performance* for information about allocating free space for inserts and updates
- The *Oracle9i Database Administrator's Guide* for more information on starting Oracle databases

# 3

# Administering Storage Components in Real Application Clusters

This chapter describes how to administer storage components in **Real Application Clusters**. It includes the following topics:

> **See Also:** *Oracle9i Real Application Clusters Deployment and Performance* for information about optimizing the use of storage components to improve performance

# Adding Datafiles in Real Application Clusters

You can add a datafile while your Real Application Clusters database is running. When you do this, Oracle automatically controls resource assignments to the new file just as it controls resource assignments for existing files.

# Automatic Undo Management In Real Application Clusters

This section explains how to use automatic undo management and manual undo management in Real Application Clusters. Oracle offers undo space management features in Real Application Clusters that augment the undo space management features available in single-instance environments.

You can use either automatic undo management or manual undo management to manage undo space. However, Oracle Corporation strongly recommends that you use the more transparent automatic undo management method. The undo space management topics in this section are:

- Using Automatic Undo Management
- Overriding Automatic Undo Management by Using Manual Undo Management

**Note:** You cannot simultaneously use automatic undo management and manual undo management mode. In other words, all instances within Real Application Cluster environments must operate in the same undo mode.

## Using Automatic Undo Management

With automatic segment-space management, Oracle manages the in-segment free and used space with bitmaps, as opposed to free lists. Automatic segment-space management is simpler to administer than free lists and it provides improved space utilization. This is especially true for objects with highly varying size rows. Automatic segment-space management also improves run-time adjustment to variations in concurrent access. Moreover, it provides improved cluster performance in terms of performance/space utilization.

To use automatic undo management, set the following parameters:

- Set the global parameter UNDO_MANAGEMENT to auto in your **server parameter file**. If you use client-side parameter files, then the setting for UNDO_MANAGEMENT must be identical in all the files.

- Set the UNDO_TABLESPACE parameter to assign undo tablespaces to instances.

You must have already created any undo tablespaces you use with the UNDO_TABLESPACE parameters. Otherwise, the STARTUP command fails.

If you do not set the UNDO_TABLESPACE parameter, then each **instance** uses the first available undo tablespace. If undo tablespaces are not available, then the instances use the SYSTEM rollback segment. Therefore, Oracle Corporation recommends that you assign an UNDO TABLESPACE to a specific instance to control their use.

**Warning:** Oracle Corporation recommends that you do *not* use the SYSTEM rollback segment for undo. When you use the SYSTEM rollback segment for undo, Oracle writes a message to the alert files to warn that your database is running without undo tablespaces.

When using automatic undo management, Oracle ignores settings for the TRANSACTIONS parameter. This is because Oracle dynamically allocates transaction objects from the System Global Area (SGA) for automatic undo management.

## Switching Undo Tablespaces

You can dynamically re-direct undo tablespace use by executing the ALTER SYSTEM SET UNDO_TABLESPACE statement. For example, assume you have instances db1 and db2 accessing undo tablespaces undotbs01 and undotbs02 respectively. If you have an idle undo tablespace, for example, undotbs03, you can execute the following statement from either instance to re-direct undo processing from that instance to undotbs03:

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

**Note:** Each instance can only use one undo tablespace at a time. In addition, instances cannot share undo tablespaces.

User transactions proceed normally while Oracle executes this operation. In some circumstances, an instance can temporarily access two undo tablespaces at the same time. This only happens while transition processing occurs during the tablespace switching operation.

This process also does not wait for all user transactions to commit. Instead, it 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.

In the last code example, the previously-used undo tablespaces, `undotbs01` or `undotbs02` remain owned by the instance until the instance's last active transaction has committed.

Although each undo tablespace can only be used by one instance at any one time, all instances can read undo blocks for consistent read purposes at any time. Also, any instance is allowed to 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.

> **See Also:** Refer to the *Oracle9i Database Administrator's Guide* for more information about the `ALTER SYSTEM SET UNDO_TABLESPACE` statement

## System Rollback Segment

When you use automatic undo management, the only external rollback segment Oracle uses is the `SYSTEM` rollback segment. There is only one `SYSTEM` rollback segment for each database. The `SYSTEM` rollback segment resides in the `SYSTEM` tablespace and Oracle automatically creates it during database creation.

In Real Application Clusters databases, all instances use the same `SYSTEM` rollback segment. Under normal circumstances, the `SYSTEM` rollback segment is only used for performing system transactions, such as the creation of transaction tables. You normally do not have to perform any operations to manage the `SYSTEM` rollback segment.

> **See Also:** The *Oracle9i Database Administrator's Guide* for information about the remaining administrative operations you can perform on undo tablespaces, such as setting the undo retention period and dropping undo tablespaces

## Public and Private Rollback Segments

Public and private rollback segments provide the same level of performance. However, private rollback segments provide more control over the matching of instances with rollback segments. This enables you to locate the rollback segments for different instances on different disks to improve performance. Therefore, use private rollback segments to improve performance.

Public rollback segments form a pool of rollback segments that can be acquired by any instance needing an additional rollback segment. Using public rollback segments can be disadvantageous, however, when you simultaneously shut down and start up instances.

For example, assume you shut down instance X and it releases public rollback segments. Then you start instance Y and it acquires the released rollback segments from instance X. Finally, you start instance and it cannot acquire its original rollback segments. Thus, instances are forced to acquire public rollback segments at startup if you do not properly set the TRANSACTIONS and TRANSACTIONS_PER_ROLLBACK_SEGMENTS parameters.

You can use public rollback segments to improve space usage. For example, if you create only one large public rollback segment for long-running transactions that run on different instances each month, then you can take the rollback segment offline and bring it back online or move it from one instance to another to better serve instances with the heavier processing demands.

By default, a rollback segment is private and is used by the instance specifying it in the parameter file. Specify private rollback segments using the ROLLBACK_SEGMENTS parameter.

In addition, the following rules also apply:

- Once a public rollback segment is acquired by an instance, it is then used exclusively by that instance.
- Once created, you can bring both public and private rollback segments online using the ALTER ROLLBACK SEGMENT command.
- If an instance has more than the SYSTEM rollback segment online, then you need at least one more rollback segment to start the instance.

**See Also:**

- *Oracle9i Database Administrator's Guide* for more information about how instances acquire rollback segments and how to monitor rollback segments
- *Oracle9i Database Reference* for a description of DBA_ROLLBACK_SEGS and DBA_SEGMENTS

## Overriding Automatic Undo Management by Using Manual Undo Management

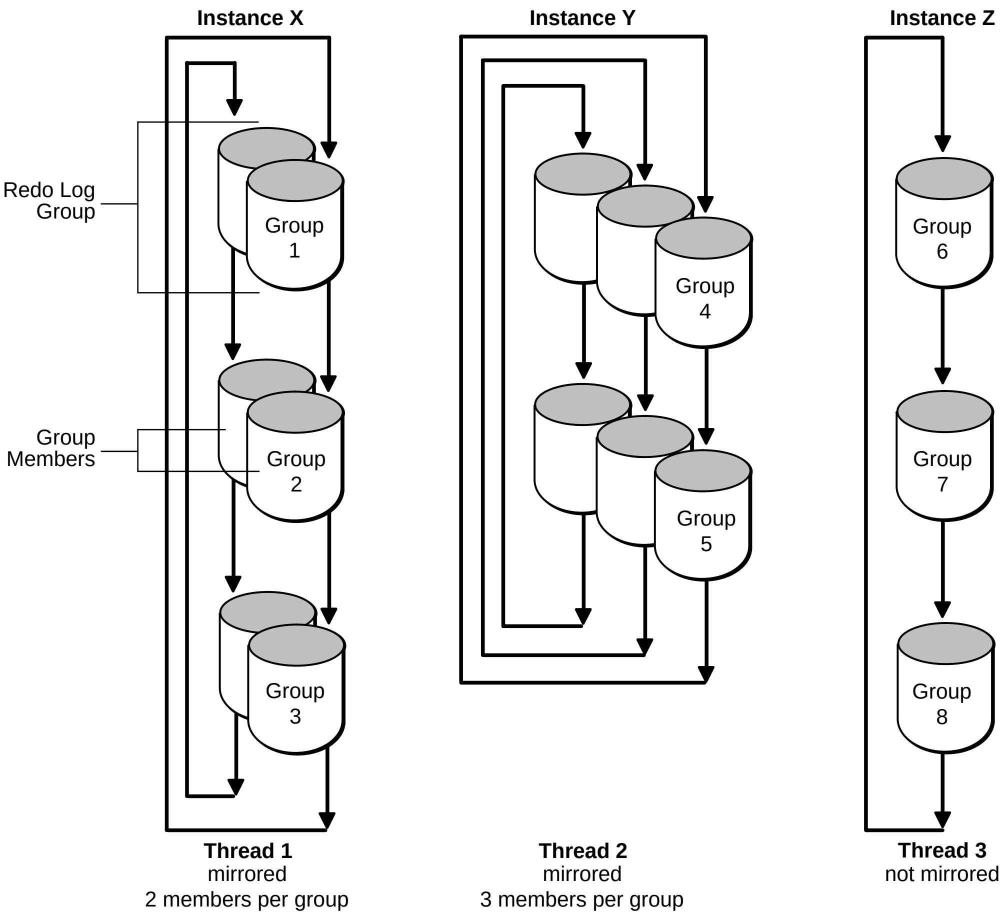
Although not recommended, you can override the default automatic undo management by setting the `UNDO_MANAGEMENT` parameter to *manual*. In addition, follow the recommendations in the rest of this section. If you do not specify the `UNDO_MANAGEMENT` parameter, then Oracle starts the instance in automatic undo management mode.

> **See Also:** Appendix B, "Associating Instances and Users with Free Lists and Free List Groups (Optional)" for more information on manual undo management and free lists

# Using Redo Log Files 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 *Oracle9i Database Administrator's Guide*. Figure 3–1 shows the threads of redo for three Real Application Clusters instances.

***Figure 3–1 Threads of Redo***



- Instance X uses thread 1 that contains three *groups* of online redo log group**s**. Thread 1 is multiplexed, that is, each group has two copies, or *members*, of the redo log file.

- Instance Y uses thread 2 that contains two groups of online redo log files. Thread 2 is multiplexed and each group of redo logs has three members.
- Instance Z uses thread 3 that contains three groups of online redo log files, but these redo log files are not multiplexed; there is only one member for each group.

---

**Note:** The **Database Configuration Assistant (DBCA)** creates two log files, one member for each group and one thread for each instance.

---

Group numbers must be unique within the database. However, the order of assigning groups to threads and threads to instances is arbitrary.

For example, although in Figure 3–1 thread 1 contains groups 1, 2, and 3, while thread 2 contains groups 4 and 5, you could instead assign groups 2, 4, and 5 to thread 1 while assigning groups 1 and 3 to thread 2. The `V$LOGFILE` view displays the group number associated with each redo log file.

Although it is possible to have different numbers of groups and members for each thread, Oracle Corporation recommends that you configure all threads using a standard that facilitates administration. That is, if possible configure all of your threads like those shown for thread X or Y. Oracle Corporation recommends against using non-mirrored redo log groups as shown for thread Z. Non-standard redo log configurations can be useful, however, for performance reasons.

Different degrees of mirroring may be required for some instances that perform better with less mirroring overhead. For example, one instance could have three groups with two members for each group, a second instance could have four non-multiplexed log files, and a third instance could have two groups with four members for each group.

In Real Application Clusters, each instance must have at least two groups of online redo log files. When the current group fills, an instance begins writing to the next log file group. At a redo log switch, Oracle writes information to the control file that identifies the filled group and its thread number after it has been archived.

---

**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.

---

**See Also:** *Oracle9i Database Administrator's Guide* for a full description of multiplexed redo log files

## Managing Trace Files and Alert Files in Real Application Clusters

Oracle records information about important events that occur in your Real Application Clusters environment in trace files and alert files. The trace files and alert files for Real Application Clusters are the same as those in single-instance Oracle databases.

Monitor these files frequently and regularly copy of them for all instances. This preserves their content and avoids accidentally overwriting the files.

**See Also:** "Using Trace Files" on page A-2 for more information about trace files and alert logs

## The Sequence Number Generator in Real Application Clusters

Real Application Clusters enables users on multiple instances to generate unique sequence numbers with minimal synchronization. The sequence number generator enables multiple instances to access and increment a sequence without contention among instances for sequence numbers and without waiting for transactions to commit.

Each instance can have its own sequence cache for faster access to sequence numbers. Oracle uses **Global Cache Service (GCS) resources** to coordinate sequences across instances in Real Application Clusters.

This section describes the `CREATE SEQUENCE` statement and its options.

- The CREATE SEQUENCE Statement
- The CACHE Option
- The ORDER Option

### The CREATE SEQUENCE Statement

The SQL statement `CREATE SEQUENCE` establishes a database object from which multiple users can generate unique integers without waiting for other users to commit transactions to access the same sequence number generator.

Real Application Clusters enables users on multiple instances to generate unique sequence numbers with minimal cooperation or contention among instances.

Sequence numbers are always unique, unless you use the CYCLE option. However, you can assign sequence numbers out of order if you use the CACHE option without the ORDER option, as described in the following section.

> **See Also:** *Oracle9i SQL Reference* for more information about the CREATE SEQUENCE and CYCLE options

## The CACHE Option

The CACHE option of CREATE SEQUENCE preallocates sequence numbers and retains them in an instance's SGA for faster access. You can specify the number of sequence numbers cached as an argument to the CACHE option. The default value is 20.

Caching sequence numbers significantly improves performance but can cause the loss of some numbers in the sequence. In other words, the sequence numbers will not be in chronological order. Losing sequence numbers is unimportant in some applications, such as when sequences are used to generate unique numbers for primary keys.

A cache for a given sequence is populated at the first request for a number from that sequence. After the last number in that cached set of numbers is assigned, the cache is repopulated with another set of numbers.

Each instance keeps its own cache of sequence numbers in memory. When an instance shuts down, cached sequence values that have not been used in committed DML statements can be lost. The potential number of lost values can be as great as the value of the CACHE option multiplied by the number of instances shutting down. Cached sequence numbers can be lost even when an instance shuts down normally.

## The ORDER Option

The ORDER option of CREATE SEQUENCE guarantees that sequence numbers are generated in the order of the requests. You can use the ORDER option for time-stamp numbers and other sequences that must indicate the request order across multiple processes and instances.

If you do not need Oracle to issue sequence numbers in order, the NOORDER option of CREATE SEQUENCE can significantly reduce overhead in a Real Application Clusters environment.

**Note:** Real Application Clusters databases do not support the `CACHE` option with the `ORDER` option of `CREATE SEQUENCE` when the database is mounted in cluster mode. Oracle cannot guarantee an order if each instance has some sequence values cached. Therefore, if you should create sequences with both the `CACHE` and `ORDER` options, they will be ordered but not cached.

# 4

# Administering Real Application Clusters Databases with the Server Control Utility, SQL, and SQL*Plus

This chapter explains how to administer instances and databases in **Real Application Clusters** environments using the **Server Control (SRVCTL) Utility**, SQL, and SQL*Plus. The topics in this chapter include:

- Administering Real Application Clusters Environments with SRVCTL
- Overview of Using SRVCTL to Administer Real Application Clusters
- Global Services Daemon (GSD)
- SRVCTL Administrative Tasks
- SRVCTL Command Syntax
- Common SRVCTL Syntax Components
- Importing and Exporting Raw Device Configurations with SRVCONFIG
- Upgrading Oracle8i Configurations to Oracle9i
- Administering Real Application Clusters Databases Using SQL and SQL*Plus
- Setting and Connecting to Instances
- Quiescing A Real Application Clusters Database
- How SQL and SQL*Plus Commands Affect Instances

**See Also:** Chapter 5, "Administering Real Application Clusters Databases with Oracle Enterprise Manager" for information on using **Oracle Enterprise Manager** to administer Real Application Clusters

## Administering Real Application Clusters Environments with SRVCTL

This chapter explains how to administer Real Application Clusters environments using Real Application Clusters with the Oracle **Server Control (SRVCTL) Utility**. This chapter includes the following topics:

- Overview of Using SRVCTL to Administer Real Application Clusters
- Global Services Daemon (GSD)
- SRVCTL Command Syntax
- Common SRVCTL Syntax Components

**See Also:** Chapter 5, "Administering Real Application Clusters Databases with Oracle Enterprise Manager" for information about using Oracle Enterprise Manager to administer Real Application Clusters

## Overview of Using SRVCTL to Administer Real Application Clusters

Oracle Corporation recommends that you use SRVCTL as the primary tool to administer the Real Application Clusters environment. SRVCTL manages configuration information that is used by several Oracle tools. For example, Oracle Enterprise Manager and **Oracle Intelligent Agent** use 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 SRVM configuration repository that is used with other Oracle administrative interfaces.

**See Also:** *Oracle9i Real Application Clusters Setup and Configuration* for more information about the SRVM configuration repository

**Note:** Before and after each configuration change, back up the SRVM configuration repository.

Before using SRVCTL, ensure that your **Global Services Daemon (GSD)** is running. Do this by executing the appropriate command to run the GSD based on your platform. Oracle should respond with a message stating that the GSD is already running. For information on GSD, Refer to "Global Services Daemon (GSD)" on page 4-3.

**Note:** To use SRVCTL, you must have already created the configuration information for the database that you want to administer. You must have done this either by using the `srvctl add` command as described in the chapter about manually creating Real Application Cluster databases, or by using the **Database Configuration Assistant (DBCA)**. Both of these topics are described in *Oracle9i Real Application Clusters Setup and Configuration*.

**See Also:** *Oracle Intelligent Agent User's Guide* for more information about the Oracle Intelligent Agent

## Global Services Daemon (GSD)

The Global Services Daemon (GSD) records information such as connection requests from SRVCTL and stores these records the `gsdaemon_`*`node_name`*`.log` file in the `$ORACLE_HOME/srvm/log` directory. Clients of the GSD, such as SRVCTL, the DBCA, and Oracle Enterprise Manager, interact with the GSD to perform various manageability operations on the nodes in your cluster.

Clients of the Global Services Daemon (GSD), such as SRVCTL, the DBCA, and Oracle Enterprise Manager, interact with the GSD to perform various manageability operations on the nodes in your cluster database. You must start the GSD on all the nodes in your Real Applications Clusters database so that the manageability features and tools operate properly.

For example, if you start an instance using Oracle Enterprise Manager, then the Intelligent Agent launches a script that contains SRVCTL commands. The GSD executes these commands which correspond to the requested operation.

### UNIX GSD Implementations

The name of the Global Services Daemon on UNIX platforms is *gsd* and is located in the `$ORACLE_HOME/bin` directory. The GSD records information such as connection requests from SRVCTL and stores these records the `gsdaemon_`*`node_name`*`.log` file in the `$ORACLE_HOME/srvm/log` directory, where *node_name* represents the instance number to support cluster file system environments.

### Windows GSD Implementations

The name of the GSD service on Windows NT and Windows 2000 platforms is *OracleGSDService* and is located in the `%ORACLE_HOME%\bin` directory. The GSD service records information such as connection requests from SRVCTL and stores these records in the `gsdaemon_`*`node_name`*`.log` file in the `%ORACLE_HOME%\srvm\log` directory, where *`node_name`* represents the name of the node to support cluster file system environments.

## SRVCTL Administrative Tasks

This section describes the tasks you can accomplish using the SRVCTL utility to administer Real Application Clusters. You can use SRVCTL to perform two types of administrative tasks as listed under the following sub-headings:

- SRVCTL Cluster Database Tasks
- SRVCTL Cluster Database Configuration Tasks

### SRVCTL Cluster Database Tasks

- Start and stop cluster databases
- Start and stop cluster database instances
- Obtain the statuses of a cluster database, cluster database instances

### SRVCTL Cluster Database Configuration Tasks

- Add and delete cluster database configuration information
- Add an instance to and delete an instance from the configuration of a cluster database
- Move instances in a cluster database configuration

- Set and unset the environment for an instance in a cluster database configuration
- Set and unset the environment for an entire cluster database in a cluster database configuration

# SRVCTL Command Syntax

This section describes the SRVCTL command syntax.

## SRVCTL Syntax Components

SRVCTL syntax has the following components:

```
srvctl verb noun options
```

Where:

- `srvctl` is the SRVCTL command.
- *`verb`* is an action word such as start, stop, or remove.
- *`noun`* is an object upon which SRVCTL performs the action verb, such as database or instance. Or you can use abbreviations, which in this case are db and inst respectively.
- *`options`* extends the use of preceding verb-noun combinations to include addition information for the command. That is, the -i option indicates that a comma-separated list of instance names will follow (sometimes the -i option only permits one value and not a list of names), or the -n option indicates that a node name or a comma-separated list of node names follows.

To see the online command syntax and options for each SRVCTL command, enter:

```
srvctl verb noun -h
```

To see a list of command verbs and nouns, enter:

```
srvctl
```

To see the version number of SRVCTL, enter:

```
srvctl -V
```

## Concurrent SRVCTL Commands

One SRVCTL action is executed on one object at a time. SRVCTL does not support concurrent execution of commands on the same object.

## Stopping SRVCTL Operations

If you enter Control-C to stop an SRVCTL operation, then the control-C is trapped. If SRVCTL displays progress messages for a command that is executing, then these messages stop and control returns to you. By default, the commands that are executing or queued to execute do not stop.

## SRVCTL Error Messages

SRVCTL error messages are documented in Appendix C, "Real Application Clusters Management Tools Error Messages" of this book.

# Common SRVCTL Syntax Components

The following are the common SRVCTL command syntax verbs and options. All operations performed with SRVCTL are case insensitive and case preserving.

## Command Syntax

The following section describes the SRVCTL command syntax. The syntax described in this section is of the format:

```
srvctl verb noun [options]
```

### Common Verbs

Table 4–1 lists the common verbs in alphabetical order. You can use these verbs with any SRVCTL command.

***Table 4–1 Common SRVCTL Verbs***

| Verb | Meaning |
|---|---|
| `add` | Add a database or instance |
| `config` | List the configuration for the database or instance |
| `getenv` | List the environment variables in the SRVM configuration |
| `modify` | Modify the instance configuration |

*Table 4–1 (Cont.) Common SRVCTL Verbs*

| Verb | Meaning |
|---|---|
| `remove` | Remove the database or instance |
| `setenv` | Set the environment variable in the SRVM configuration |
| `start` | Start the database or instance |
| `status` | Status of the database or instance |
| `stop` | Stop the database or instance |
| `unsetenv` | Set the environment variable in the SRVM configuration to unspecified |

## Common Nouns

Table 4–2 lists the common nouns in alphabetical order. You can use these nouns with any SRVCTL command.

*Table 4–2 Common SRVCTL Nouns*

| Noun (Abbreviation) | Meaning |
|---|---|
| `database (db)` | Operation refers to objects for the database |
| `instance (inst)` | Operation refers to objects for the instances |

## Syntax Flags

Table 4–3 shows the various SRVCTL command syntax flags and their meanings.

*Table 4–3 SRVCTL Syntax Flags*

| Flag | Meaning |
|---|---|
| -h | Help |
| -i | instance |
| -n | node |
| -f | force |

## Common Options

Table 4–4 lists the common command options in alphabetical order. You can use these options with any SRVCTL command. Other options depend on the verb that is being executed.

*Table 4–4 Common SRVCTL Command Options*

| Option | Meaning |
|---|---|
| `-d` | Database name |
| `-h` | Print usage |
| `-i` | Comma-separated list of instance names for the operation |
| `-n` | Node name or comma separated node list |

## SRVCTL Commands

The following section describes the SRVCTL commands in alphabetical order.

### SRVCTL Add

Adds configuration information for the database or for named instances. When adding an instance, the name that you specify with -i should match the `INSTANCE_NAME` and `ORACLE_SID` parameters.

```
srvctl add database -d database_name [-m domain_name] -o oracle_home [-s spfile]
srvctl add instance -d  database_name  -i instance_name -n node_name
```

*Table 4–5 Command-Specific Options for SRVCTL Add*

| Command | Option |
|---|---|
| -m | Database domain name, in the form “us.mydomain.com”. |
| The database domain name for the `-m` option must match the `DB_DOMAIN` and `DB_NAME` parameters in `INIT.ORA` or `SPFILE`. When adding a database, the name that you specify with `-d` should match the `DB_NAME` parameter. | |
| n | Node name that will support an instance. |
| -o | `$ORACLE_HOME` to locate lsnrctl (node option) and Oracle binaries (other options). |
| -s | SPFILE name. |

### Examples of SRVCTL Add

The following are examples of using the `add` command.

To add a new database:

```
srvctl add  database  -d mydb -o /ora/ora9
```

To add named instances to a database:

```
srvctl add instance -d mydb -i  mydb01 -n gm01
srvctl add instance -d mydb -i  mydb02 -n gm02
srvctl add instance -d mydb -i  mydb03 -n gm03
```

### SRVCTL Config

Displays the configuration that is stored in the SRVM configuration file.

```
srvctl config database
```

Displays a list of configured databases.

```
srvctl config database -d database_name
```

Displays the specified database configuration in the following format:

```
nodename1 instancename1 oraclehome
nodename2 instancename2 oraclehome
```

### Examples of SRVCTL Config

The following are examples of using the `config` command.

To display database configuration:

```
srvctl config database -d mydb
```

### SRVCTL Getenv

The `getenv` operation gets and displays values for the environment from the SRVM configuration file.

```
srvctl getenv database -d database_name [-t name[,name,...]]
srvctl getenv instance -d database_name -i instance_name [-t name[,name,...]]
```

### Example of SRVCTL Getenv

To list all environment variables for a database:

```
srvctl getenv database -d mydb
```

## SRVCTL Modify

Modify enables modification of the instance-node configuration. Using `modify` preserves the environment in the SRVM configuration that would otherwise need to be re-entered. The configuration description is modified in the SRVM configuration repository. The change takes effect when the application is next restarted.

Modify with the instance option moves the named instance to a new node. This node cannot support another instance for the same database. The move is permanent.

```
srvctl modify instance -d database_name -i instance_name -n node_name
```

## Example of SRVCTL Modify

The following is an example of using the `modify` command.

To modify a named instance to execute on another node:

```
srvctl modify  instance  -d mydb  -n my_new_node
```

## SRVCTL Remove

This command removes the configuration information from the SRVM repository. Environment settings for the object are also removed. If you do not use the force flag (-f), then Oracle prompts you to confirm the removal.

You are then asked if you wish to proceed. With the force (-f) option, the remove proceeds without prompting.

```
srvctl remove database -d database_name [-f]
srvctl remove instance  -d database_name -i instance_name [-f]
```

## Command-Specific Options for SRVCTL Remove

| | |
|---|---|
| -f | Force the removal of the application without prompting. |

## Examples of SRVCTL Remove

The following are examples of using the `remove` command.

To remove the applications for a database:

```
srvctl remove  database  -d mydb
```

To remove the applications for named instances of a database:

```
srvctl remove instance -d mydb -i  mydb01
srvctl remove instance -d mydb -i  mydb02
srvctl remove instance -d mydb -i  mydb03
```

## SRVCTL Setenv

The `setenv` operation sets values for the environment in the SRVM configuration file.

```
srvctl setenv database -d database_name -t name=value [,name=value,…]
srvctl setenv instance -d database_name [-i instance_name] -t name=value
[,name=value,…]
```

## Example of SRVCTL Setenv

The following is an example of using the `setenv` command.

Set environment for database:

```
srvctl setenv database -d mydb -t LANG=en
```

## SRVCTL Start

Starts the database, all or named instances, and all listeners associated with the database if they are not already started.

> **Note:** For the `start` command, and for other operations that use a connect string, if you do not provide a connect string, then Oracle uses "/ as sysdba" to perform the operation on the instance. In other words, to execute such operations you must be a member of the OSDBA group. For more information on OSDBA group membership, refer to the *Oracle9i Installation Guide*.

```
srvctl start database -d database_name [-o  start_options] [-c connect_string]
srvctl start instance -d database_name -i instance_name [,instance_name-list]
[-o start_options] [-c connect_string]
```

***Table 4–6 Command-Specific Options for SRVCTL Start***

| Command | Option |
|---|---|
| `-o` | Options passed directly to startup command in SQL*Plus including PFILE. |
| `-c` | Connect string for connecting to the Oracle instance using SQL*Plus. |

### Examples of SRVCTL Start

The following are examples of the `start` command.

To start the database and all enabled instances:

```
srvctl start database -d mydb
```

To start instances:

```
srvctl start instance  -d mydb -i mydb1,mydb4
```

### SRVCTL Status

Display the current state of the named database.

```
srvctl status database -d database_name
srvctl status instance -d database_name -i instance_name [,instance_name-list]
```

### Examples of SRVCTL Status

The following are examples of using the `status` command.

Status of the database and all instances:

```
srvctl status database -d mydb
```

Status of named instances:

```
srvctl status instance -d mydb -i mydb1,mydb2
```

## SRVCTL Stop

Stops the database and all or named instances. This does not stop listeners.

```
srvctl stop database -d database_name [-o stop_options] [-c connect_string]
srvctl stop instance -d database_name -i instance_name [,instance_name_list] [-o
stop_options][-c connect_string]
```

*Table 4–7 Command-Specific Options for SRVCTL Stop*

| Command | Option |
|---|---|
| -c | Connect string for connecting to the Oracle instance using SQL*Plus |
| -o | Options passed directly to shutdown command in SQL*Plus |

### Examples of SRVCTL Stop

The following are examples of using the `stop` command.

Stop the database all instances:

```
srvctl stop database -d mydb
```

Stop named instances:

```
srvctl stop instance  -d mydb -i mydb1
```

## SRVCTL Unsetenv

The `unsetenv` operation unsets values for the environment in the SRVM configuration file.

```
srvctl unsetenv database -d database_name -t name[,name,...]
srvctl unsetenv instance -d database_name [-i instance_name] -t name[,name,...]
```

### Example of SRVCTL Unsetenv

The following is an example of using the `unsetenv` command.

Unset an environment variable back to unspecified:

```
srvctl unsetenv database -d mydb -t CLASSPATH
```

## Importing and Exporting Raw Device Configurations with SRVCONFIG

You can use `SRVCONFIG` to import and export raw device configuration information whether the configuration file resides on a cluster file system file or on a raw device. You might do this to back up or restore the SRVM configuration information. For example, the following syntax exports the contents of the configuration information to the text file that you name:

```
srvconfig -exp file_name
```

As another example, the following imports the configuration information from the text file you name to the configuration repository for the Real Application Clusters environment in which you execute the command:

```
srvconfig -imp file_name
```

## Upgrading Oracle8*i* Configurations to Oracle9*i*

If you are upgrading from Oracle8*i* to Oracle9*i*, upgrade your configuration information using the following post-installation procedure. Do this for each Real Application Clusters database:

1. Stop all Global Services Daemons (GSD) by executing the `gsdctl stop` command.
2. Execute the following command from the node in a UNIX cluster where the *db_name*`.conf` file is located:

```
srvconfig -conv $Oracle_Home/ops/db_name.conf
```

### Administering the GSD

You can use *gsdctl* commands to start, stop, and obtain the status of the GSD service on any platform. The options for `gsdctl` are:

- `gsdctl start` — To start the GSD service
- `gsdctl stop` — To stop the GSD service
- `gsdctl stat` — To obtain the status of the GSD service

# Administering Real Application Clusters Databases Using SQL and SQL*Plus

Although Oracle Corporation recommends that you use SRVCTL to administer your Real Application Clusters database environment, you can also use SQL and SQL*Plus. Prior to performing the tasks with SQL and SQL*Plus as described in this section, ensure your **Cluster Manager (CM)** component is started on each node.

The SQL and SQL*Plus procedures you use to start your cluster database depends on your platform as explained under the following sub-headings:

- Starting Databases in Cluster Mode on UNIX
- Starting Databases in Cluster Mode on Windows NT and Windows 2000

## Starting Databases in Cluster Mode on UNIX

To start a Real Application Clusters database in cluster mode:

1. Ensure your Cluster Manager software is running. Instructions on Cluster Manager software administration appear in your operating system-specific documentation. If the Cluster Manager is not available or if Oracle cannot communicate with this component, Oracle displays the error ORA-29701: "Unable to connect to Cluster Manager".

2. Start any required operating system-specific processes. For more information about these processes, see your operating system-specific documentation.

3. If the **listener** is not started, start it on each of the nodes. Enter:

   ```
   LSNRCTL
   LSNRCTL> start [listener_name]
   ```

   Where *listener_name* is the name of the listener defined in the **listener.ora** file. It is not necessary to identify the listener if you are using the default listener named LISTENER.

   LSNRCTL displays a status message indicating that the listener started successfully. You can check that all expected services for that listener are listed in the services summary in the status message. You can also check the status of the listener with the LSNRCTL STATUS command.

4. Start the database on one of the nodes by starting SQL*Plus. Then enter:

   ```
   CONNECT SYS/password as SYSDBA
   STARTUP PFILE=init$ORACLE_sid.ora
   ```

The first instance to start in cluster mode determines the values of any global parameters for the other instances. When another instance attempts to start in cluster mode, the Real Application Clusters database compares the values of any global parameters in its parameter file with those already in use and issues messages if any values are incompatible. An instance cannot mount the database unless it has the correct values for its global parameters.

5. On the remaining nodes, start the database:

```
CONNECT SYS/password as SYSDBA
STARTUP PFILE=$ORACLE_sid.ora;
```

## Starting Databases in Cluster Mode on Windows NT and Windows 2000

To start the Real Application Clusters database in cluster mode on Windows platforms:

1. Start OracleService*sid* instance on *each* node.

   - From the MS-DOS command line enter:

     ```
     C:\> net start OracleServicesid
     ```

   - From the Control Panel's Services window, select OracleServicesid, then click Start.

2. If the listener is not started, start it on each of the nodes. Enter:

   ```
   LSNRCTL
   LSNRCTL> start [listener_name]
   ```

   Where *listener_name* is the name of the listener defined in the `listener.ora` file. You do not have to identify the listener if you are using the default listener named `LISTENER`.

   `LSNRCTL` displays a status message indicating that the listener started successfully. You can check that all expected services for that listener appear in the services summary in the status message. You can also check the status of the listener with the `LSNRCTL STATUS` command.

3. Start the database on one of the nodes by starting SQL*Plus. Then enter:

   ```
   CONNECT SYS\password
   STARTUP PFILE=%ORACLE_HOME%\database\initsid.ora;
   ```

The first instance to start in cluster mode determines the values of any global parameters for the other instances. When another instance attempts to start in cluster mode, the Real Application Clusters database compares the values of any global parameters in its parameter file with those already in use and issues messages if any values are incompatible. The instance cannot mount the database unless it has the correct values for its global parameters.

4. On the remaining nodes, start the database:

```
CONNECT SYS\password
STARTUP PFILE=%ORACLE_HOME%\database\initsid.ora;
```

## Using RETRY to Mount a Database in Cluster Mode

If you attempt to start an instance and mount a database in cluster mode while another instance is recovering the same database, your current instance cannot mount the database until the recovery is complete. Rather than repeatedly attempting to start the instance, use the `STARTUP RETRY` statement. This causes the new instance to retry mounting the database every five seconds until it succeeds or has reached the retry limit. Use the syntax:

```
STARTUP OPEN database_name RETRY
```

To set the maximum number of times the instance attempts to mount the database, use the SQL*Plus `SET` command with the `RETRY` option. You can specify either an integer such as `10`, or the keyword `INFINITE`.

If the database can only be opened by being recovered by another instance, then using the `RETRY` does not repeat connection attempts. For example, if the database was mounted in exclusive mode by one instance, then trying the `STARTUP RETRY` command in cluster mode does not work for another instance.

> **Note:** Because an instance startup does not affect the datafiles, you can start an instance without mounting the datafiles.

# Setting and Connecting to Instances

Before setting instances and connecting to them, you must install and configure Oracle Net for the Real Application Clusters nodes and any clients that access these nodes. This establishes remote connections from the clients to the nodes.

**See Also:**

- *Oracle9i Real Application Clusters Setup and Configuration*
- *Oracle9i Net Services Administrator's Guide*

SQL*Plus commands operate on the current instance with some exceptions as noted under the next heading, "The SET INSTANCE and SHOW INSTANCE Commands" on page 4-19.

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. Because the SQL*Plus prompt does not show which instance is the current instance, be sure you 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 switch the current instance from the local instance to a remote instance, do one of the following:

- Re-execute the `CONNECT` command specifying a remote instance **net service name** as in this example:

```
CONNECT SYSTEM/MANAGER@net_service_name
```

- Disconnect from the database and execute a `SET INSTANCE` command as in this example:

```
SET INSTANCE net_service_name
```

Issue another `CONNECT` command with only your user ID and password. Specifying a remote instance with the `CONNECT` command while connected to the database by way of an instance enables you to switch from one instance to another without disconnecting.

**See Also:**

- The *Oracle9i Net Services Administrator's Guide* for information on configuring net service names
- Your operating system-specific Oracle documentation for more information about the exact format required for the connect string used in the `SET INSTANCE` and `CONNECT` commands

## The SET INSTANCE and SHOW INSTANCE Commands

When using SET INSTANCE to specify an instance on a remote node for the STARTUP command, the parameter file for the remote instance must be accessible by the local node.

The SHOW INSTANCE command displays the net service name for the current instance. SHOW INSTANCE returns the value local if you have not used SET INSTANCE during the SQL*Plus session.

To reset to the default instance, use SET INSTANCE without specifying a net service name or specify local. Do not follow the SET INSTANCE command with the word default; this syntax specifies a connect string for an instance named default.

## The CONNECT Command

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:**

- *The Oracle9i Net Services Administrator's Guide* or the proper specification of *net_service_name*
- *Oracle9i Database Administrator's Guide* for information on connecting to the database using SYSDBA or SYSOPER privileges

## Verifying That Instances are Running

To verify that instances are running:

1. On any node, enter:

```
CONNECT SYS/password
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
```

Where the output columns from this SELECT statement are as described in Table 4–8:

***Table 4–8 Descriptions of V$ACTIVE_INSTANCES Columns***

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

## Shutting Down Real Application Clusters Instances

Shutting down Real Application Clusters instances is procedurally identical to shutting down instances in single instance environments with these exceptions:

- In Real Application Clusters, shutting down one instance does not interfere with the operation of other running instances.
- To shut down a Real Application Clusters database mounted in shared mode, shut down every instance in the Real Application Clusters environment.
- When you shut down an instance abnormally, Oracle forces all user processes running in that instance to log off the database. If a user process is currently accessing the database, Oracle terminates that access and displays the message "ORA-1092: Oracle instance terminated. Disconnection forced". If a user process is not currently accessing the database when the instance shuts down, Oracle displays the message "ORA-1012: Not logged on" upon the next call or request made to Oracle.
- 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 SMON process of an 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.

- If multiple SQL*Plus sessions are connected to the same instance simultaneously, all but one must disconnect before the instance can shut down normally. You can use the IMMEDIATE or ABORT option of the SHUTDOWN command to shut down an instance when multiple SQL*Plus sessions (or any other sessions) are connected to it.

> **See Also:** *Oracle9i Database Administrator's Guide* for more information on shutting down Oracle databases

## Quiescing A Real Application Clusters Database

Quiescing a Real Application Clusters database is procedurally identical to quiescing a single-instance database except as described in this section. For example, you cannot open the database on one instance if the database is *being quiesced*. In other words, if you issued the ALTER SYSTEM QUIESCE RESTRICTED statement but Oracle has not finished processing it, you cannot open the database. Nor can you open the database if it is already in a quiesced state. In addition, the ALTER SYSTEM QUIESCE RESTRICTED and ALTER SYSTEM UNQUIESCE statements affect all instances in a Real Application Clusters environment, not just the instance that issues the command.

> **See Also:** The *Oracle9i Database Administrator's Guide* for details on the quiesce database feature and the *Oracle9i SQL Reference* for more information about the ALTER SYSTEM QUIESCE RESTRICTED syntax

## How SQL and SQL*Plus Commands Affect Instances

Most SQL statements affect the current instance. For example, 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 also affects the current instance. By contrast, ALTER SYSTEM CHECKPOINT or ALTER SYSTEM CHECKPOINT GLOBAL affects *all* instances.

ALTER SYSTEM SWITCH LOGFILE affects only the current instance. To force a global log switch, use the ALTER SYSTEM ARCHIVE LOG CURRENT statement. The THREAD option of ALTER SYSTEM ARCHIVE LOG enables you to archive each online redo log file for a specific instance.

Table 4–9 describes how common SQL*Plus commands affect instances.

***Table 4–9 How SQL*Plus Commands Affect Instances***

| SQL*Plus Command | Associated Instance |
|---|---|
| ARCHIVE LOG | Always affects to the current instance. |
| CONNECT | Affects the default instance if no instance is specified in the CONNECT command. |
| HOST | Affects to 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 | Display parameter and SGA information from the current instance. |
| STARTUP and SHUTDOWN | Always affects the current instance. These are privileged SQL*Plus commands. |

**Note:** The security mechanism that Oracle uses when you execute privileged SQL*Plus commands depends on your operating system. Most operating systems have a secure authentication mechanism when logging onto the operating system. On these systems, your default operating system privileges usually determine whether you can use STARTUP and SHUTDOWN. For more information, refer to your operating system-specific documentation.

# Part II

# Using Oracle Enterprise Manager to Administer Real Application Clusters

Part II contains information about using the Server Management (SRVM) component of Oracle Enterprise Manager. The chapter in Part II is:

# 5

# Administering Real Application Clusters Databases with Oracle Enterprise Manager

This chapter describes how to use **Oracle Enterprise Manager** to administer Real Application Clusters databases. This chapter only describes Oracle Enterprise Manager administration for Real Application Clusters. Use this section as a supplement to information in the *Oracle Enterprise Manager Administrator's Guide* and other Oracle Enterprise Manager documentation. The topics in this chapter are:

- Overview of Oracle Enterprise Manager Administration
- Starting the Console
- Displaying Objects in the Navigator Pane
- Using the Cluster Database Right-Mouse Menu
- Starting a Cluster Database
- Shutting Down a Cluster Database
- Viewing the Cluster Database Operation Results
- Viewing Cluster Database Status
- Creating a Job for a Cluster Database or Instance
- Specifying Job Details
- Registering Cluster Database Events

## Overview of Oracle Enterprise Manager Administration

The Oracle Enterprise Manager **Console** provides a central point of control for the Oracle environment through a graphical user interface (GUI). You can use the Oracle Enterprise Manager Console to initiate a variety of cluster database management tasks.

Once you have created or modified the configuration of a cluster database, then you should use Oracle Enterprise Manager to discover the nodes on which you configured the cluster database. Once discovery completes, you can administer all of the databases and their instances and listeners that are associated with the nodes discovered by Oracle Enterprise Manager.

This enables you to start, stop, and monitor databases, 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.

> **See Also:** *Oracle9i Real Application Clusters Setup and Configuration* for configuration information and the Oracle Enterprise Manager online help system for answers to frequently asked questions

## Starting the Console

To use the Oracle Enterprise Manager Console, start the following components:

1. An **Oracle Intelligent Agent** on each node
2. A **Global Services Daemon (GSD)** on each node
3. The **Management Server**
4. The Oracle Enterprise Manager Console

> **See Also:** *Oracle9i Real Application Clusters Setup and Configuration* for Real Application Clusters-specific information on installing Oracle Enterprise Manager and *Oracle Intelligent Agent User’s Guide* for more information about the Oracle Intelligent Agent

## Displaying Objects in the Navigator Pane

From the Navigator pane, you can view and manage cluster database instances in the same manner as you view and manage single-instance databases. After Oracle Enterprise Manager discovers a node, the Navigator displays all the cluster databases, their instances and their listeners that are discovered on that node.

The information available for cluster databases is essentially the same as for single-instance databases. Just as in single-instance databases, cluster databases and their related elements can be administered using master and detail views of the Oracle Enterprise Manager Console.

In the Navigator tree, cluster databases are listed under the Databases folder as siblings to single-instance databases. Just as in single-instance databases, each cluster database folder contains the instances and sub-folders for Instance, Schema, Security, Storage, and Distributed as shown in Figure 5–1 on page 5-5.

> **Note:** If you attempt to expand a cluster database and you have not yet set preferred credentials for the database as described in *Oracle9i Real Application Clusters Setup and Configuration*, then the Database Connect Information dialog prompts you to enter database connect information.

By selecting objects within a cluster database subfolder, you can access property sheets to inspect and modify properties of these objects, just as you can for single-instance databases. For example, by expanding the Storage folder and then right-clicking Redo Log Groups subfolder and then choosing **Create**, you can add a new redo log group. Oracle Enterprise Manager displays all discovered cluster database instances under the Cluster Database Instances folder.

When accessing cluster databases, only the Instance folder has subfolder contents that differ from those seen in single-instance databases. Within the Instance folder, the instance database subfolders are split into two parts:

- Database-Specific Objects
- Instance-Specific Objects

## Database-Specific Objects

All of the database-specific functionality appears directly under the Instance subfolder for the cluster database. The available database-specific functionality enables you to view, create, and modify resource consumer groups.

## Instance-Specific Objects

All instance-specific functionality appears beneath the individual instance icons within the Cluster Database Instances subfolder. The instance-specific functionality includes:

### Configuration

You can view instance states, view and edit initialization parameters, toggle the archive mode on and off, and view performance statistics of active resource plans. You can also view or modify the current undo tablespace assigned to an instance, as well as the undo tablespace retention period.

### Stored Configurations

You can create, edit, and store multiple startup configurations for instances. This eliminates the need to track `init`*`sid`*`.ora` parameter files.

### Sessions

You can list the status of connected users, view the latest SQL for sessions, and terminate sessions.

### Locks

You can view details for currently held User type and System type locks.

### Resource Plans

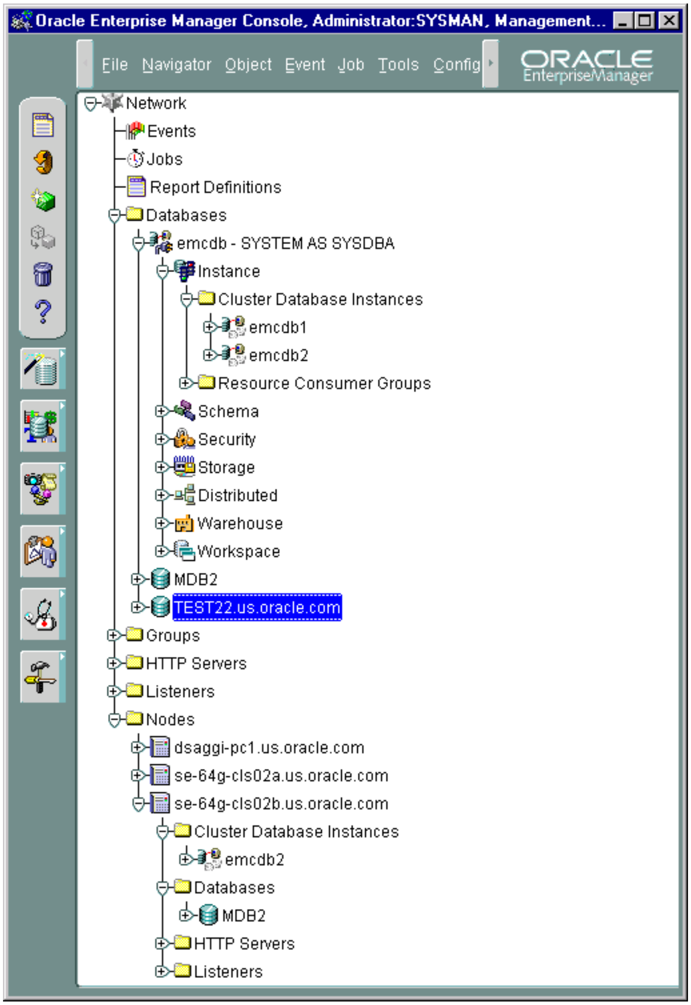
You can define and modify the resource plans for the cluster database and also activate a specific resource plan for an individual instance.

### Resource Plan Schedule

You can schedule the activation of a resource plan.

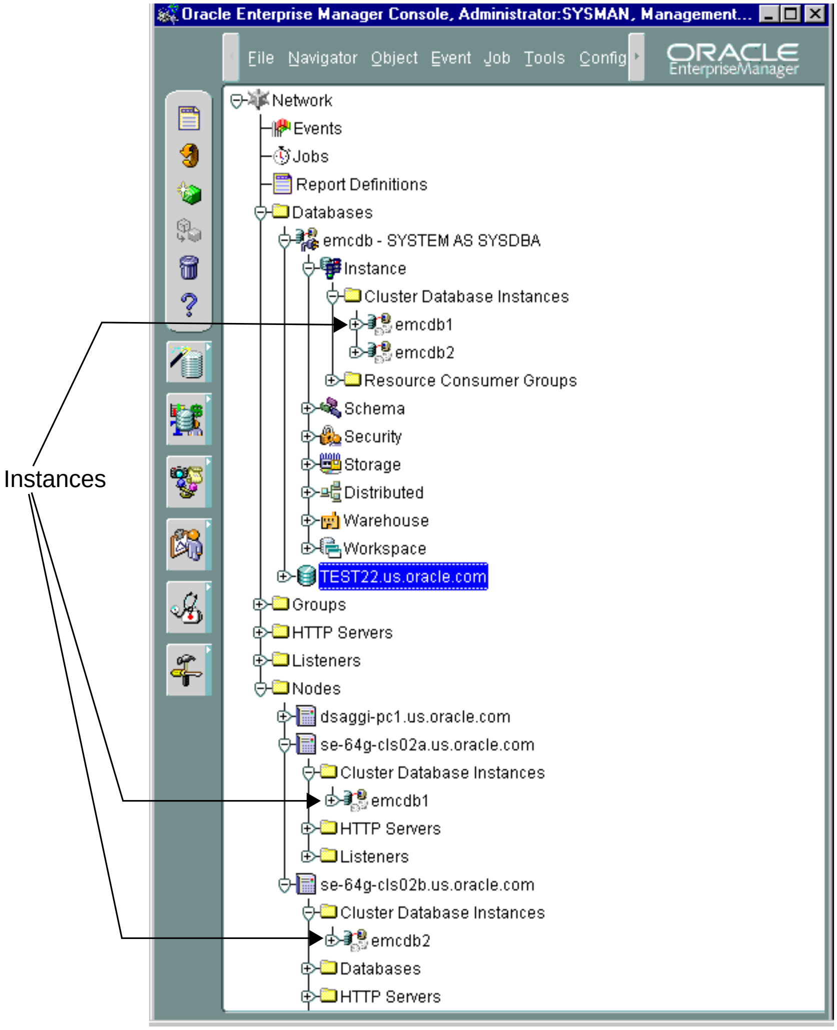
The contents of a typical database folder are shown in Figure 5–1.

***Figure 5–1 Database Subfolders***



You can expand the Cluster Database Instances folder to display the instances belonging to each discovered database or node, as shown in Figure 5–2.

***Figure 5–2 Cluster Database Instances Folder***



---

**Note:** The instances have the same right-mouse menu as the right-mouse menu used in single-instance databases.

---

## Using the Cluster Database Right-Mouse Menu

Right-clicking on a cluster database displays the specialized cluster database right-mouse menu. The right-mouse menu contains entries for the functions listed in Table 5–1. All other menus are identical to those appearing in the single-instance version of Oracle Enterprise Manager. When selected, you will see dialogs that differ from single-instance databases.

*Table 5–1 Right-Mouse Menu Functions for Real Application Clusters*

| Option | Description |
|---|---|
| Startup | Starts the database<br>**See Also:** "Starting a Cluster Database" on page 5-7 |
| Shutdown | Shuts down the database<br>**See Also:** "Shutting Down a Cluster Database" on page 5-9 |
| Results | Displays startup and shutdown results<br>**See Also:** "Viewing the Cluster Database Operation Results" on page 5-10. |
| View Edit Details | Enables inspection of the state of the cluster database, including which instances are active.<br>**See Also:** "Viewing Cluster Database Status" on page 5-13 |
| Related Tools | Contains access to other tools which have been enabled for a cluster database. |

## Starting a Cluster Database

The Console enables you to start an entire cluster database or selected instances within a cluster database. To start a cluster database or its instances:

1. In the Navigator pane expand Databases.
2. Right-click a cluster database.

   A menu appears with options for the database.
3. Choose **Startup** from the menu.

   The Cluster Database Startup dialog box appears.
4. Select a startup type as shown in Table 5–2.

*Table 5–2 Cluster Database Startup Types*

| Option | Description |
|---|---|
| No Mount | Does not mount the database upon instance startup |
| Mount | Mounts a database but does not open it |
| Open | (*default*) Opens the database |
| Force the instance(s) to start | Shuts down any currently running Oracle instances with the SHUTDOWN mode, ABORT, before restarting them. If the instances are running and FORCE is not specified, an error results.<br>**Note:** You should *not* use the FORCE mode under normal circumstances. Use the FORCE mode only while debugging or under abnormal circumstances. |
| Restrict access to the database | Makes the started instances accessible only to users with the RESTRICTED SESSION system privilege. Users already connected are not affected. |

5. To start up all instances and their related listeners, click **Startup**. To start up only selected instances, follow these steps:

   a. Click **Select Instances**.

      The Select Instances to Start dialog box appears.

   b. Select the instances to start in the Available list and click **Add**.

   c. Click **OK** to close the Select Instances to Start dialog box.

   d. Click **Startup** from the Cluster Database Startup dialog box.

   The Cluster Database Startup Results dialog box displays the progress of the startup operation, as described in "Viewing the Cluster Database Operation Results" on page 5-10. If the instances were started successfully, the Cluster Database Started message box appears with a successful message. Click **OK** in the Cluster Database Started message to acknowledge the message, then click **Close** in the Cluster Database Startup Results.

   If the startup fails, the Cluster Database Started message box appears with a failure message. Click **View Details** to see more information in the Cluster Database Startup Results dialog box about the failure, then click **Close**.

## Shutting Down a Cluster Database

The Console enables you to shut down an entire cluster database or selected instances within a cluster database. Once all instances are shut down, the database is also considered shut down.

**Note:** A cluster database can occasionally be completely down, however, some of its services, such as the database listener, might remain running.

To shut down a cluster database or its instances:

1. In the Navigator pane expand Databases.
2. Right-click a cluster database.

   A menu appears with options for the database.
3. Choose **Shutdown** from the menu.

   The Cluster Database Shutdown dialog box appears.
4. Select a shutdown type, as shown in Table 5–3.

***Table 5–3 Shutdown Types***

| Option | Description |
|---|---|
| Normal | Waits for the currently connected users to disconnect from the database, prohibits further connects, and closes and dismounts the database before shutting down the instance. Instance recovery is not required on next startup. |
| Immediate | *(default)* Does not wait for current calls to complete, prohibits further connects, and closes and dismounts the database.The instance is immediately shut down. Connected users are not required to disconnect and instance recovery is not required on next startup. |
| Abort | Proceeds with the fastest possible shutdown. Connected users are not required to disconnect. The database is not closed or dismounted, but the instances are shut down. Instance recovery is required on next startup.<br>**Note:** You must use this option if a background process terminates abnormally. |

| Option (Cont.) | Description |
|---|---|
| Transactional | Waits for transactions to complete before shutting down |
| Shutdown Database Only | (*default*) Shuts down the database only. The services required for an instance, such as the listener, remain up and available. |
| Shutdown Database And Other Services | Shuts down the database and associated services, such as the listener. |

5. To shut down all instances, click **Shutdown**.

   To shut down only selected instances, follow these additional steps:

   a. Click **Select Instances**.

      The Select Instances to Stop dialog box appears.

   b. Select the instances to stop in the Available list, then click **Add**.

   c. Click **OK** to close the Select Instances to Stop dialog box.

   d. Click **Shutdown** from the Cluster Database Shutdown dialog box.

   The Cluster Database Shutdown Progress dialog box displays the progress of the shutdown operation.

   If the instances were shut down successfully, the Cluster Database Stopped message box appears with a successful message. Click **OK** in the Cluster Database Stopped message to acknowledge the message, then click **Close** in the Cluster Database Shutdown Results window.

   If the shutdown fails, the Cluster Database Stopped message box appears with a failure message. Click **View Details** to see information in the Cluster Database Shutdown Progress dialog box about the shutdown failure, then click **Close**.

## Viewing the Cluster Database Operation Results

The Cluster Database Startup / Shutdown Results dialog displays information about the progress of the instance startup or shutdown operation you selected: The operation results are presented in two views:

- Status Details Tab
- Output Tab

The Cluster Database Startup / Shutdown Results dialog box automatically displays during a startup or shutdown operation. You can also initiate it with the following steps:

1. In the Navigator pane, expand Databases.
2. Right-click a cluster database.

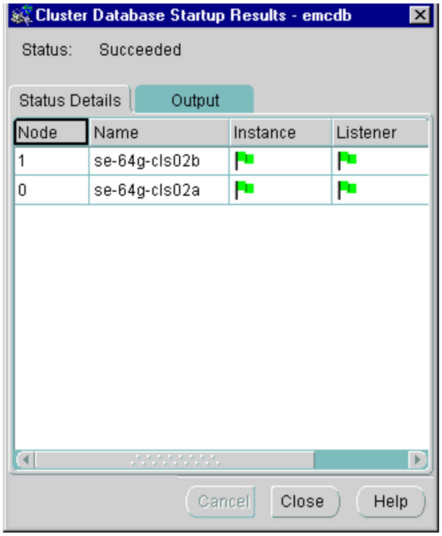
   A menu appears with options for the database.
3. Choose **Results** from the menu.

## Status Details Tab

While a startup or shutdown operation is running against a cluster, the Status Details tab appears and is updated dynamically as the operation progresses.

A successful startup operation for a two-node cluster looks like the following in the Status Details tab, as shown in Figure 5–3.

***Figure 5–3 Status Details Tab***



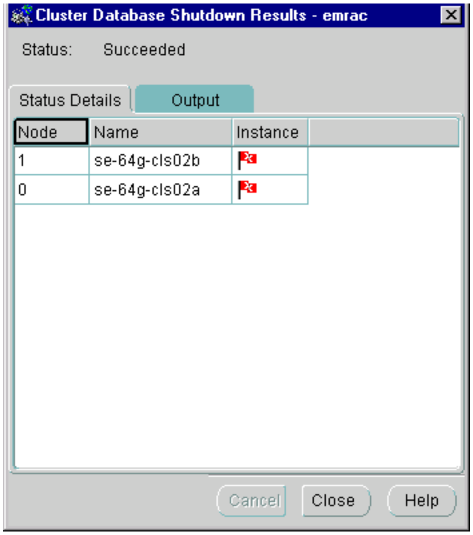
Cluster Database Startup Results - emcdb

Status: Succeeded

| Node | Name | Instance | Listener |
|---|---|---|---|
| 1 | se-64g-cls02b | ⚑ | ⚑ |
| 0 | se-64g-cls02a | ⚑ | ⚑ |

A successful shutdown operation for a two-node cluster looks like the Status Details tab shown in Figure 5–4.

*Figure 5–4 Successful Shutdown Results*



Cluster Database Shutdown Results - emrac

Status: Succeeded

| Node | Name | Instance |
|---|---|---|
| 1 | se-64g-cls02b | 🚩 |
| 0 | se-64g-cls02a | 🚩 |

Table 5–5 shows the possible states in which each component might appear.

*Table 5–4 Possible Component States*

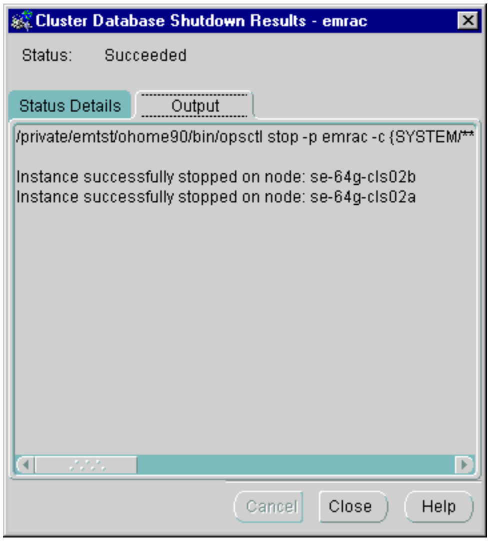
| State | Description |
|---|---|
| Up (green flag) | The component is running. |
| Down (red flag) | The component is not running. |
| In Progress (timer) | Oracle Enterprise Manager cannot determine the state of the component. This state occurs typically when the component startup or shutdown operation has not completed. |
| Component does not exist on this node (blank background) | The component was not configured on the node.<br>Not all components (listener, instance) are required on every node. |

## Output Tab

The Output tab displays the commands executed by the node and any associated error messages in textual format.

A successful shutdown looks like the following in the Output tab as shown in Figure 5–5.

*Figure 5–5 Successful Shutdown Results*



Status: Succeeded

```
/private/emtst/ohome90/bin/opsctl stop -p emrac -c {SYSTEM/**

Instance successfully stopped on node: se-64g-cls02b
Instance successfully stopped on node: se-64g-cls02a
```

Oracle Enterprise Manager displays a Results tab that is similar to the Shutdown tab in Figure 5–5.

# Viewing Cluster Database Status

The Edit Cluster Database dialog box displays status information about the database, such as instances available in the cluster and the status of cluster components.

---

**Note:** Because this dialog box requires a connection to a database, this dialog box will not appear if the cluster database is down.

---

To view status information about a database:

1. In the Navigator pane, expand Databases > *database_name*.
2. Right-click a cluster database under the Databases folder in the Navigator pane.

   A menu appears with options for the database.
3. Choose **View/Edit Details** from the menu.

   The Edit Cluster Database dialog box appears.

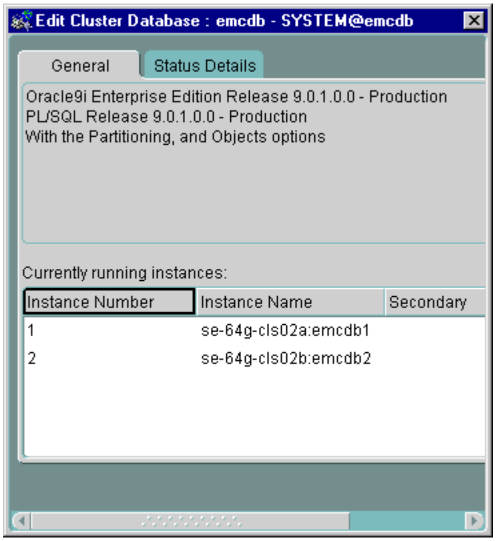
The operation results are presented in two views:

- General Tab
- Status Details Tab

## General Tab

The General tab displays information about the currently running instances by querying the `V$ACTIVE_INSTANCES` table as shown in Figure 5–6. Oracle Enterprise Manager makes a connection to the cluster database. Therefore, this tab will not appear if the database is down.

*Figure 5–6 General Tab*



Oracle9i Enterprise Edition Release 9.0.1.0.0 - Production
PL/SQL Release 9.0.1.0.0 - Production
With the Partitioning, and Objects options

Currently running instances:

| Instance Number | Instance Name | Secondary |
|---|---|---|
| 1 | se-64g-cls02a:emcdb1 | |
| 2 | se-64g-cls02b:emcdb2 | |

Table 5–5 describes the Fields in Output Tab Form.

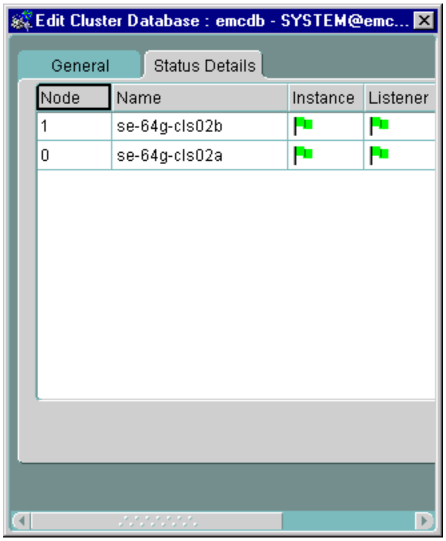
*Table 5–5 Fields in Output Tab Form*

| Column Name | Description |
|---|---|
| Instance Number | Identifies the **instance number** |
| Instance Name | The name specified for the instance and the node it is running on. This name has the following format: *node*:*instance_name*. |
| Secondary | A checkmark appears if the node is an secondary instance in a primary/secondary instance configuration |

## Status Details Tab

The Status Details Tab displays an overall view of the state of the cluster and related components as shown in Figure 5–7. This tab displays the status of the various components, such as listeners and instances, for all nodes on which you have configured the cluster database.

*Figure 5–7 Status Details Tab*



| Node | Name | Instance | Listener |
|---|---|---|---|
| 1 | se-64g-cls02b | ⚑ | ⚑ |
| 0 | se-64g-cls02a | ⚑ | ⚑ |

Table 5–4 shows the possible states in which each component might appear.

# Creating a Job for a Cluster Database or Instance

The job scheduling system provides a highly reliable and flexible mechanism for you to schedule and automate repetitive jobs on both the cluster database and instances.

The Console contains a full-featured scheduling mechanism that enables you to develop a customized schedule. This provides you with actual “lights out” management capability so that you can focus on other tasks.

You can create a job with a cluster database or instance as the target. To create a new job, follow these steps:

1. Choose **Job** > **Create Job**.
2. Complete the tabs of the Create Job property sheet.
3. Click **Submit** to submit the job to Oracle Intelligent Agent. The job appears in the Active Jobs window.
4. Select **Submit** to submit the job, **Add to Library** to add the job to the library, or **Submit and Add** to do both. Then click the button on the lower right-hand side relative to the operation you want to perform. The job appears in the Job Library window. You can later modify or submit a saved job.

# Specifying Job Details

From the Create Job property sheet, you can specify the details of a new job. The Create Job property sheet contains the tabs listed in Table 5–6.

***Table 5–6 Tabs Contained in the Create Job Property Sheet***

| Tab | Description |
|---|---|
| General | Specify the job name, description, target type, and destination. |
| Tasks | Choose the task(s) for the job. |
| Parameters | Set the run-time parameters for the tasks. The parameters that appear on this tab depend on which task(s) you chose on the Task list box. |
| Schedule | Schedule the time and frequency for Oracle Enterprise Manager to run the job. |
| Permissions | Specify the administrator to perform the job. |

The following tabs contain options specific to cluster databases:

- General Tab
- Tasks Tab
- Parameters Tab

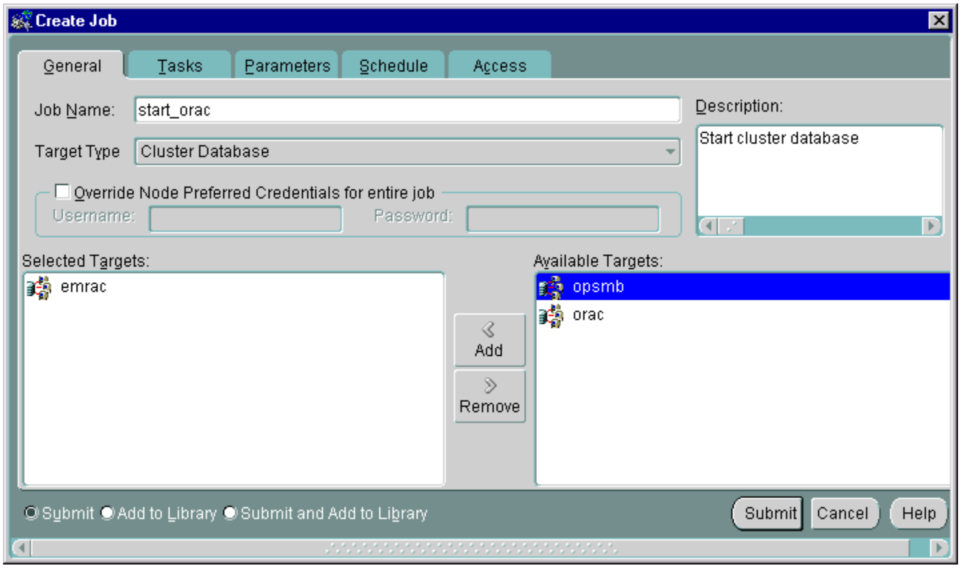
> **See Also:** *Oracle Enterprise Manager Administrator's Guide* for general job scheduling information

## General Tab

As shown in Figure 5–8, from the General tab you can specify:

- Job Name
- Target Type (either Cluster Database or Cluster Database Instance)
- Description
- Targets

***Figure 5–8 General Tab***



Job Name: start_orac

Target Type: Cluster Database

☐ Override Node Preferred Credentials for entire job

Username:

Password:

Description: Start cluster database

Selected Targets: emrac

Available Targets: opsmb, orac

○ Submit ○ Add to Library ○ Submit and Add to Library

The General Tab contains the options in Table 5–7.

***Table 5–7 General Tab Options***

| Option | Description |
|---|---|
| Job Name | Enter the name of the new job. |
| Description | Enter a description of the job. |
| Target Type | Select a Target Type from the drop-down list box. You can select from the following options: Cluster Database, Cluster Database Instance, Database, Listener, or Node. |
| Available Targets | The targets are determined by your selection of the Target Type. The targets include Cluster Databases, Cluster Database Instances, Databases, Listeners, and Nodes.<br>Click the targets of the job in the Available Targets list, then click Add to move the target to the Selected Targets list. To remove a target from a job, click the target in the Selected Targets list, then click **Remove**. |

## Tasks Tab

From the Tasks tab, choose the task(s) for the job to perform. The list of tasks that appears is different depending on whether you select a cluster database or cluster database instance as your Target Type from the General tab. Move the tasks between the Available Tasks and Job Tasks lists with the Add and Remove buttons.

### Tasks for Cluster Database Targets

If your Target Type is a Cluster Database, then you can select from these tasks that are specific to cluster databases:

- Shutdown Cluster Database
- Startup Cluster Database

> **See Also:** *Oracle Enterprise Manager Administrator's Guide* for a description of these tasks and the parameters to set

## Parameters Tab

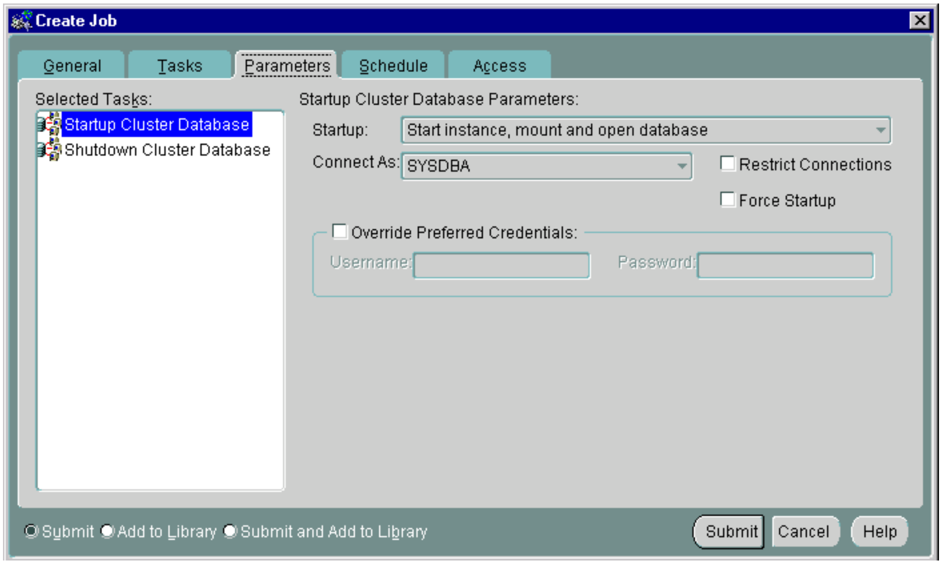
From the Parameters tab, you can specify parameter settings for the job tasks you selected on the Tasks tab. The parameters that display vary according to the job task. Parameters specific to cluster database startup and shutdown tasks are described in the next subsection.

> **See Also:** *Oracle Enterprise Manager Administrator's Guide* for a description of parameters to set for instance tasks

### Cluster Database Startup Task

When you select the **Startup Cluster Database** task on the Tasks tab, the following display appears Figure 5–9.

***Figure 5–9 Parameters Tab***



Selected Tasks:
- Startup Cluster Database
- Shutdown Cluster Database

Startup Cluster Database Parameters:

Startup: Start instance, mount and open database

Connect As: SYSDBA

□ Restrict Connections

□ Force Startup

□ Override Preferred Credentials:

Username:

Password:

○ Submit ○ Add to Library ○ Submit and Add to Library

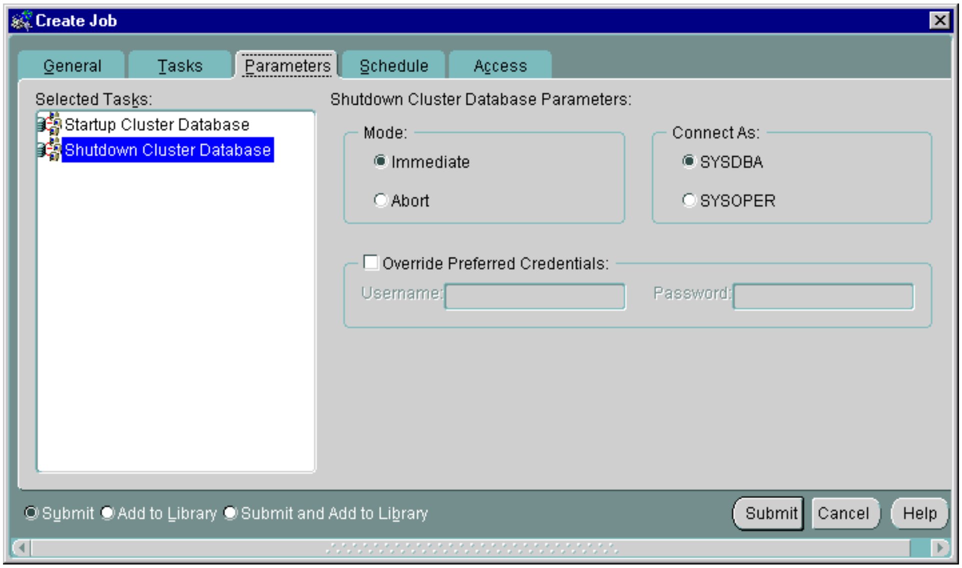
The Parameters tab for Startup contains the options listed in Table 5–8.

***Table 5–8 Parameters Tab for Startup***

| Parameters | Description |
|---|---|
| Startup | Select the startup modes for the job from the drop-down list box. |
| Connect As | Specify the role.<br>Only Normal is allowed for Oracle7. For Oracle8*i* and subsequent releases, **SYSOPER** and **SYSDBA** roles allow you the maximum database administration privileges. You require SYSDBA or SYSOPER privileges to run job tasks such as shutdown or startup on the database.<br>**See Also**: *Oracle9i Database Administrator's Guide* for more information about SYSOPER and SYSDBA roles<br>If you attempt to connect as SYSDBA and do not have SYSDBA privileges, an error message states that an invalid user name or password was entered. |
| Override Preferred Credentials | You can use the preferred credentials that have been set up for the database, or you can enter another database user name and password. |

## Parameters for the Cluster Database Shutdown Task

When you select **Shutdown Cluster Database** on the Tasks tab, the display in Figure 5–10 appears.



Create Job

Selected Tasks:
Startup Cluster Database
Shutdown Cluster Database

Shutdown Cluster Database Parameters:

Mode:
● Immediate
○ Abort

Connect As:
● SYSDBA
○ SYSOPER

□ Override Preferred Credentials:
Username:
Password:

○ Submit ○ Add to Library ● Submit and Add to Library

***Figure 5–10 Shutdown Cluster Database Parameters***

Complete the parameter entries on the tab and click **Submit** to run the cluster database shutdown task. The Parameters tab for Shutdown contains the options listed in Table 5–9.

***Table 5–9 Parameters Tab for Shutdown***

| Options | Description |
|---|---|
| Mode | Select **Immediate** (*default*) or **Abort** |
| Connect As | Select **SYSDBA** (*default*) or **SYSOPER**. |
| Override Preferred Credentials | You can use the preferred credentials that have been set up for the database or you can use another database user name and password. |

## Registering Cluster Database Events

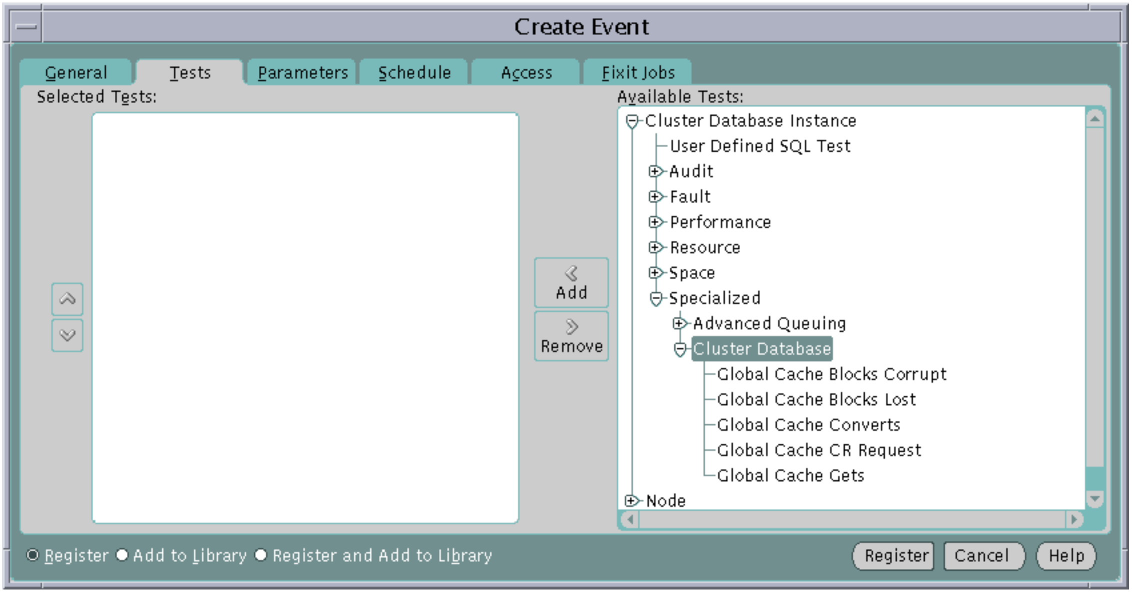
Oracle Enterprise Manager includes an Event Management feature that enables you to monitor tests on cluster database instances. You can run all tests available for single-instance databases on cluster database instances. In addition to these standard event tests, specialized event tests are available only for cluster database instances. These tests include global cache converts, consistent read requests, and so on. The Intelligent Agents running on the cluster nodes gather the data for the event tests and detect the occurrence of specific event conditions.

To access the Event Management feature, from the Event menu, select **Create Event** and the Create Event window opens. On the Create Event window you can select targets, set parameters, and set notification preferences, using the following sub-pages:

- Use the General Page to select the cluster database and cluster database instance target types.
- Use the Test Page to select the type of specialized cluster database tests that you want to schedule.
- Use the Parameters Page to select the parameter for your test event.
- Use the Schedule Page to schedule the execution of a test event.
- Use the Access Page for setting notification options.
- Use the Fixit Jobs Page to specify a job that should be run after the event occurs.

Figure 5–11 shows the selection list of event tests that are specific to cluster databases.

*Figure 5–11 Event Management Tests Menu*



# Part III

# Backup and Recovery in Real Application Clusters

Part III explains how to use Recovery Manager (RMAN) in Real Application Clusters environments. It also explains how to perform backup and recovery in Real Application Clusters environments. The chapters in Part III are:

- Chapter 6, "Using Recovery Manager in Real Application Clusters Environments"
- Chapter 7, "Performing Backup and Recovery in Real Application Clusters Environments"

# 6

# Using Recovery Manager in Real Application Clusters Environments

This chapter explains how to use **Recovery Manager (RMAN)** in Real Application Clusters environments. The topics in this chapter include:

# Configuring RMAN in Real Application Clusters: Overview

Perform the procedures described in this chapter after installing the **Oracle Enterprise Edition** with **Oracle Real Application Clusters** and after reviewing the following documentation:

- *Oracle9i Recovery Manager User's Guide*
- *Oracle9i Recovery Manager Reference*

> **See Also:** *Oracle Enterprise Manager Administrator's Guide* to use **Oracle Enterprise Manager** for backup and recovery

The configuration procedures described in this chapter explain how to:

1. Configure the snapshot control file location.
2. Configure the control file autobackup feature.
3. Configure the archiving scheme.
4. Change the archiving mode of the database (optional)
5. Monitor the archiver processes.

# Configuring the RMAN Snapshot Control File Location

In Real Application Clusters, identical copies of the **snapshot control file** must exist on every node that performs backups. Therefore, ensure that the same destination directory for a snapshot control file exists on all nodes used for backups.

RMAN creates a snapshot control file to obtain a consistent record of the contents of the target control file before making a backup. RMAN creates the snapshot control file on the **node** that performs backups. Run the following RMAN command to determine the default location of the snapshot control file:

```
SHOW SNAPSHOT CONTROLFILE NAME;
```

If needed, you can change the default location. For example, on UNIX 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 '?/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 that are in effect across RMAN sessions. Therefore, you do not need to execute this command again unless you want to change the location of the snapshot control file.

> **Note:** You can also specify a cluster file system (CFS) 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 Real Application Clusters.

> **See Also:** *Oracle9i Recovery Manager User's Guide* for more information about configuring the snapshot control file

## Configuring the RMAN Control File Autobackup Feature

The RMAN control file autobackup feature automatically creates backups of the control file after executing the `BACKUP` and `COPY` commands. This feature is important in disaster recovery because RMAN can restore the control file even without a recovery catalog.

If you set `CONFIGURE CONTROLFILE AUTOBACKUP` to `ON`, then RMAN automatically creates a control file and a server parameter file backup after you run the `BACKUP` or `COPY` commands. 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 on all nodes that perform 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.

> **See Also:** *Oracle9i Recovery Manager User's Guide* for more information about using the control file autobackup feature

# Managing Archived Logs in Real Application Clusters Environments

After configuring the snapshot control file location and enabling the control file autobackup feature, decide how to configure your environment to manage the archived logs that each node generates.

When a cluster node generates an archived 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 log filenames in the recovery catalog when a resynchronization occurs.

The archived 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 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 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 log backups or all nodes perform archived log backups, you need to ensure that all logs are backed up. Because only one instance can perform recovery, the node performing recovery must have read access to all logs in your cluster.

## Archiving in a Real Application Cluster Environment

Your primary configuration task is to ensure that all archived redo logs can be read from every node during recovery, and if possible during backups. This scenario illustrates the archived log naming issues for configuring archiving in your cluster database.

In this scenario, assume that the following conditions are met:

- You configure each node to write to a local archiving directory that is named the same on each node.
- You 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).
- You do not use NFS or mapped drives to enable the nodes in the cluster to gain read / write access to one another.

Configure the initialization parameters file as:

```
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 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 Real Application Clusters 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. You can avoid this problem by configuring your cluster according to the schemes described in the following section.

# RMAN Archiving Configuration Schemes

The two configuration schemes in this chapter describe a three-node UNIX system for a Real Application Clusters database. For both schemes, 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.

In Real Application Clusters, set the same value for the LOG_ARCHIVE_FORMAT parameter on all instances. You can specify a different value of LOG_ARCHIVE_DEST_*n* during recovery if the archived redo log files are not at their original archiving destinations.

This section contains the following topics:

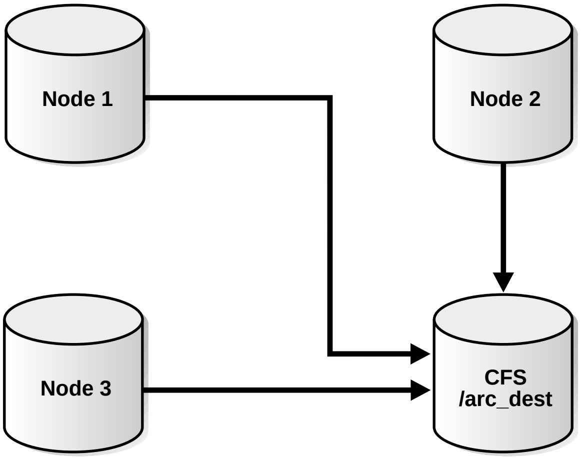
- Cluster File System Archiving Scheme
- Non-CFS Local Archiving Scheme
- Archived Log File Format and Destinations in Real Application Clusters

## Cluster File System Archiving Scheme

The preferred configuration for Real Application Clusters is to use a cluster file system archiving scheme. As shown in Figure 6–1, each node writes to a single CFS archived log destination and can read the archived log files of the other nodes. Read access is achieved with a cluster file system. For example, if node 1 archives a log to `/arc_dest/log_1_100.arc` on the CFS, then any other node in the cluster can also read this file.

**Note:** In UNIX environments only, if you do not use a cluster file system, then the archived log files cannot be on the shared disk because UNIX shared disks are raw devices that you cannot easily partition for use with archived logs.

***Figure 6–1 Cluster File System Archiving Scheme***



### Advantages and Disadvantages of the Cluster File System Archiving Scheme

The advantages of this scheme are:

- None of the nodes uses the network to archive logs.
- If each node has a local tape drive, then you can distribute an archived 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 logs.

### Initialization Parameter Settings for the Cluster File System Archiving Scheme

In the CFS 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 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/arc_1_999.arc
/arc_dest/arc_1_1000.arc
/arc_dest/arc_1_1001.arc <- thread 1 archived in node 3
/arc_dest/arc_3_1563.arc <- thread 3 archived in node 2
/arc_dest/arc_2_753.arc <- thread 2 archived in node 1
/arc_dest/arc_2_754.arc
/arc_dest/arc_3_1564.arc
```

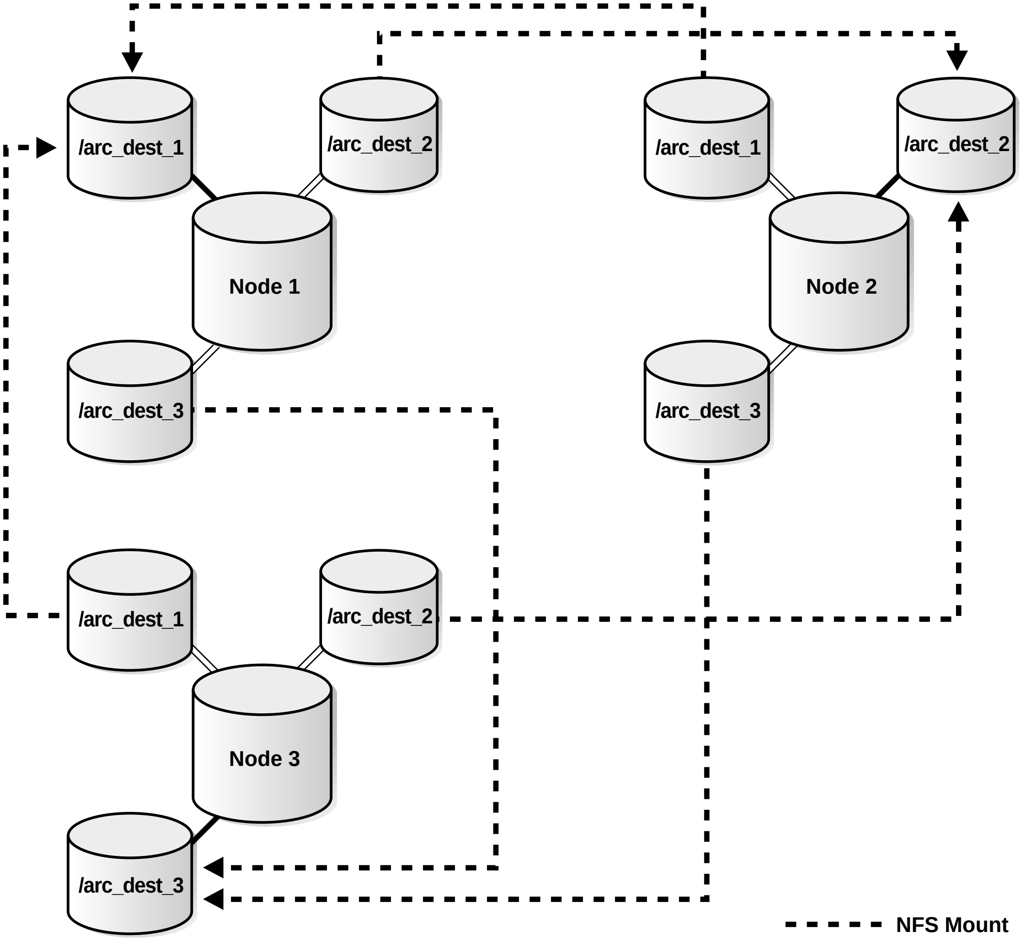
### Location of Archived Logs for the Cluster File System Archiving Scheme

Because the file system is shared and because each node is writing its archived 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-CFS Local Archiving Scheme

In the non-CFS local archiving scheme, each node archives to a uniquely named local directory as shown in Figure 6–2. 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 or shared drives on Windows platforms. Therefore, each node writes only to a local destination, but each node can also read archived log files in remote directories on the other nodes.

*Figure 6–2 Non-CFS Local Archiving Scheme*



To enable RMAN to back up and recover a Real Application Clusters database in one step, all archived 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`.

**Note:** In UNIX environments only, the archived log files cannot be on the shared disk because UNIX shared disks are raw devices that you cannot easily partition for use with archived logs.

### Advantages and Disadvantages of the Non-CFS Local Archiving Scheme

The advantage of this scheme is:

- If each node has a local tape drive, then you can distribute an archived log backup so that each node backs up local logs without accessing the network.

The disadvantages of this scheme are:

- In a media recovery scenario, you must configure the node performing recovery for remote access so that it can read the archived 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 logs on this node that were generated *after* the backup are lost. If you are in a recovery situation and do not have all available logs, then you are forced to perform point-in-time recovery up to the first missing archived log sequence number.

**Note:** 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, "Performing Backup and Recovery in Real Application Clusters Environments".

### Initialization Parameter Settings for the Non-CFS Local Archiving Scheme

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

```
/arc_dest_1/arc_1_1000.arc
/arc_dest_2/arc_1_1001.arc <- thread 1 archived in node 2
/arc_dest_2/arc_3_1563.arc <- thread 3 archived in node 2
/arc_dest_1/arc_2_753.arc <- thread 2 archived in node 1
/arc_dest_2/arc_2_754.arc
/arc_dest_3/arc_3_1564.arc
```

## Location of Archived Logs in the Non-CFS Local Archiving Scheme

As illustrated in Table 6–1, each node has a directory containing the locally archived 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 log files that are archived by the remaining nodes in the cluster.

*Table 6–1 Location of Logs in the Non-CFS Local Archiving Scheme*

| Node ... | Can read the archived 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 the Non-CFS Local Archiving Scheme

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 6–2.

***Table 6–2 NFS Configuration for Shared Read Local Archiving Scheme***

| 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 |

## Archived Log File Format and Destinations in Real Application Clusters

For any archived 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. `LOG_ARCHIVE_FORMAT` can also have variables as shown in Table 6–3. Examples in Table 6–3 assume that `LOG_ARCHIVE_FORMAT=log_%`*parameter* and that the length limits for sequence values is ten characters and thread values is four characters.

***Table 6–3 Archived Redo Log Filename Format Parameters***

| Parameter | Description | Example |
|---|---|---|
| `%T` | Thread number, left-zero-padded | `log_0001` |
| `%t` | Thread number, not padded | `log_1` |
| `%S` | Log sequence number, left-zero-padded | `log_0000000251` |
| `%s` | Log sequence number, not padded | `log_251` |

The thread parameters `%t` or `%T` are mandatory for Real Application Clusters. 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:**

- *Oracle9i Database Administrator's Guide* for information about specifying the archived redo log filename format and destination
- Oracle platform-specific documentation for information about the default log archiving format and destination

## Changing the Archiving Mode in Real Application Clusters

After configuring your Real Application Clusters environment for RMAN, you can alter the archiving mode if needed. For example, if your Real Application Clusters database uses `NOARCHIVELOG` mode, then follow these steps to change the archiving mode to `ARCHIVELOG` mode:

1. Shut down all instances.
2. Reset the `CLUSTER_DATABASE` parameter to `false` on one instance. If you are using the **server parameter file**, then make a *sid*-specific entry for this.
3. Add settings in the parameter file for the `LOG_ARCHIVE_DEST_`*n*, `LOG_ARCHIVE_FORMAT`, and `LOG_ARCHIVE_START` parameters. You can multiplex the destination to use up to ten locations. The `LOG_ARCHIVE_FORMAT` parameter should contain the `%t` parameter to include the thread number in the archived log file name. You must configure an archiving scheme before setting these parameter values.
4. Start the instance on which you have set `CLUSTER_DATABASE` to `false`.
5. Run the following statement in SQL*Plus:

   ```
   SQL>ALTER DATABASE ARCHIVELOG;
   ```

6. Shut down the instance.
7. Change the value of the `CLUSTER_DATABASE` parameter to `true`.
8. Restart your instances.

You can also change the archiving mode from `ARCHIVELOG` to `NOARCHIVELOG`. To disable archiving, follow the preceding steps with the following changes:

1. Delete the archiving settings that you created in step 3.
2. Specify the NOARCHIVELOG keyword in step 5:

   ```
   ALTER DATABASE NOARCHIVELOG;
   ```

## Querying the Archiver Processes

After your RMAN configuration is operative in your Real Application Clusters environment, use the GV$ARCHIVE_PROCESSES and V$ARCHIVE_PROCESSES views to determine the statuses of the archiver processes. Depending on whether you query the global or local views, these views display information for all database instances in your Real Application Clusters database, or for only the instance to which you are connected.

> **See Also:** *Oracle9i Database Reference* for more information about these views

# 7

# Performing Backup and Recovery in Real Application Clusters Environments

This chapter explains how to perform backup and recovery in Real Application Clusters. This chapter also includes information about using **Recovery Manager (RMAN)** for backup and recovery processing in Real Application Clusters environments. The topics in this chapter include:

- Overview of Making Backups in Real Application Clusters
- RMAN Backup Schemes for Real Application Clusters
- Instance Recovery in Real Application Clusters
- Overview of Media Recovery in Real Application Clusters
- RMAN Restore Schemes for Real Application Clusters
- Parallel Recovery in Real Application Clusters

# Overview of Making Backups in Real Application Clusters

The procedures for executing RMAN backups in Real Application Clusters environments do not differ substantially from the RMAN backup procedures for single-instance environments. However, the following topics describe a few issues that are specific to Real Application Clusters:

- Connection to the Target Database in a Real Application Clusters Configuration
- RMAN and Oracle Net in Real Application Clusters
- Channel Connections to Cluster Instances
- Node Affinity Awareness
- Accessibility of Files and Backup Media

> **See Also:** *Oracle9i Recovery Manager User's Guide* for more information about single-instance RMAN backup procedures

## Connection to the Target Database in a Real Application Clusters Configuration

When you start RMAN and connect to the target database, RMAN can only connect to one instance in a Real Application Clusters database at a time. Note that this connection is a utility connection that does not perform any backups or restores, and applies only to the connection made from the RMAN command line.

Assume that node1, node2, and node3 are net service names for three instances in a Real Application Clusters configuration. In this case, connect to the target database with only one of these net service names. For example, you can connect as follows:

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

## RMAN and Oracle Net in Real Application Clusters

In any RMAN connection made through a net service name, each net service name must specify one and only one instance. This rule applies to all RMAN connections whether they are made from the command line or through the `CONNECT` clause in `ALLOCATE CHANNEL` or `CONFIGURE CHANNEL` commands. Therefore, you cannot specify a net service name that uses Oracle Net features to distribute RMAN connections to more than one instance.

## Channel Connections to Cluster Instances

When making backups in a Real Application Clusters configuration, each allocated channel can connect to a different instance in the cluster, and each channel connection must resolve to one and 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';
```

If the instance to which one of the channels is connected does not have the database open, then the database must not be open by any instance. In other words, either all channels must be connected to open instances, or all channels must be connected to instances that are not 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

In some **cluster database** configurations, some nodes of the cluster have faster access to some datafiles than to other datafiles. RMAN automatically detects this affinity, which is known as node affinity awareness.

When deciding which channel to use to back up a particular datafile, RMAN gives preference to channels allocated at the nodes that have affinity to the datafiles 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’;
```

You can manually override the automatic node affinity by specifying which channels should back up which datafiles. For example:

```
BACKUP
  # channel 1 gets datafile 1
  (DATAFILE 1 CHANNEL ORA_SBT_TAPE_1)
```

```
# channel 2 gets datafiles 2-4
(DATAFILE 2,3,4 CHANNEL ORA_SBT_TAPE_2)
# channel 3 gets datafiles 5-10
(DATAFILE 5,6,7,8,9,10 CHANNEL ORA_SBT_TAPE_3);
```

> **See Also:** *Oracle9i Recovery Manager User's Guide* for more information about the `CONNECT` clause of the `CONFIGURE CHANNEL` command

## Accessibility of Files and Backup Media

Other important considerations for performing RMAN backups in a cluster are discussed under the following headings:

- Readability of Files To Be Backed Up
- Deleting Logs After A Successful Backup
- Distribution of Backup Media

### 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 and archived logs. 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 logs that the local node cannot read. The archiving schemes in "RMAN Archiving Configuration Schemes" on page 6-5 explain how to configure the environment so that all logs are accessible by the node performing the backup.

### Deleting Logs After A Successful Backup

The `BACKUP` command must be able to delete the archived logs from disk after backing them up. For example, the `DELETE INPUT` clause specifies that RMAN should delete only the specific log that it backed up, whereas the `DELETE ALL INPUT` clause specifies that RMAN should delete all logs that have the same thread and sequence number as the log backed up. If you are using the "Non-CFS Local

Archiving Scheme" described on page 6-8, then you can specify either DELETE INPUT or DELETE ALL INPUT.

This scheme describes an archiving scheme in which each node has read / write access to its local archiving directory, and either no read access to the others (when NFS is not set up) or read-only access to others (when the remote directories are mounted NFS read). In such cases, the best practice is *not* to specify the DELETE ALL INPUT or DELETE INPUT clauses on the BACKUP command. Instead, use the DELETE command.

The following script is an example of one method for deleting the archived 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 Backup Media

When configuring the backup media in a Real Application Clusters configuration, you have three options:

- **Network Backup Server:** None of the nodes have local drives. Instead, backups are transmitted over the network to a backup server.
- **One Local Drive**: In which only one node has access to a local tape drive
- **Multiple Drives**: In which each cluster node has access to a local tape drive

If only one node has a tape drive attached, then this node must be able to read all datafiles and archived logs. Both the "Cluster File System Archiving Scheme" on page 6-5 and the "Non-CFS Local Archiving Scheme" on page 6-8 describe scenarios in which one node in the cluster can back up all the files. However, the non-CFS scheme requires you to configure NFS so that one node can back up all of the logs.

For this reason, Oracle Corporation does not recommend that you use the non-CFS archiving scheme if you have only one local drive.

Alternatively, you can use a multiple drive scheme in which each node can write to its own local tape drive. In the CFS scheme, any node can back up all datafiles and archived logs. In the non-CFS scheme, you must write the backup script so that the backup is **distributed**. In a distributed backup, different files are sent to the drive attached to each node. For example, node 1 can back up the logs whose path names begin with `/arc_dest_1`, node 2 can back up the logs whose path names begin with `/arc_dest_2`, and node 3 can back up the 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.

The autolocation feature is important for backups of archived redo logs. If you use the "Non-CFS Local Archiving Scheme" described on page 6-8, then each node can read only a subset of all of the logs that were generated. For example, node 1 can only read logs whose path names begin with `/arc_dest_1`, node 2 can only read logs whose path names begin with `/arc_dest_2`, and node 3 can only read logs whose path names begin with `/arc_dest_3`. RMAN never attempts to back up logs on a channel unless RMAN can read the logs by using that channel. Each channel restricts its archived log backup to the logs that it is able to read.

During a restore operation, RMAN automatically performs the autolocation of backups. In other words, 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 Schemes for Real Application Clusters

This section describes the following backup schemes:

- Cluster File System Backup Scheme
- Non-CFS Backup Scheme

## Cluster File System Backup Scheme

This scheme refers to the archiving scenario described under the heading "Cluster File System Archiving Scheme" on page 6-5. In a CFS backup scheme, each node in the cluster has read access to all the datafiles and archived redo logs.

### Backing Up to One Local Drive in the CFS Archiving Scheme

This scheme assumes that only one node in the cluster has a local tape drive. In this case, execute 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 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 and archived logs.

### Backing Up to Multiple Drives in the CFS 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. 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.

**Note:** As mentioned, this is a one-time configuration step: you do not need to issue these configuration commands for every backup.

The following backup script, which you can run from any node in the cluster, distributes the datafile and archived log backups among the tape drives:

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

For example, if the database contains 10 datafiles and 100 logs are on disk, then the node 1 tape 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 tape drive can back up datafiles 4, 6, 8 and 9 as well as logs 67-100.

## Non-CFS Backup Scheme

This scheme refers to the archiving scenario described under the heading "Non-CFS Local Archiving Scheme" on page 6-8. In a non-CFS backup scheme, the datafiles are on shared disk and are accessible by all the nodes in your cluster database. Therefore, any node can back up all the datafiles. In contrast, in a non-CFS environment, each node can back up only its own local logs, that is, node 1 cannot access the logs on node 2 or node 3 unless you configure NFS for remote access.

To configure NFS, distribute the backup to multiple drives if you do not want to configure NFS for backups. However, if you configure NFS for backups, then you can only back up to one drive.

### Backing Up to Multiple Drives in a Non-CFS 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.

> **Note:** As mentioned, this is a one-time configuration step: you do not need to issue these configuration commands for every backup.

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 logs that are located on that node. The following example uses automatic channels to make a database and archived log backup:

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

In this example, the datafile backups and logs 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 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 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.

### Backing Up to One Local Drive in a Non-CFS Archiving Scheme

This scheme assumes that only one node in the cluster has a local tape drive. To make backups in this scheme, you must configure NFS so that the backup node has read access to the logs archived locally on the other nodes. For this reason, Oracle Corporation does not recommend that you back up to one local drive in a non-CFS archiving scheme. In this case, you can execute the following one-time configuration commands:

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

Because the node making the backup can read the logs archived by all of the nodes through NFS, the backup scripts do not differ from the scripts in a single-instance database. You can run the same script no matter which node is performing the backup. However, only one archiving directory on each node has read / write access. Therefore, you cannot specify `DELETE ALL INPUT` or `DELETE INPUT`. You must instead execute `DELETE` commands on each node to delete the redundant logs from disk.

For example, a production script for whole database and archived log backups from the backup node is:

```
BACKUP DATABASE PLUS ARCHIVELOG; # do not specify DELETE ... INPUT
```

To back up only the archived logs, you can run the following script:

```
BACKUP ARCHIVELOG ALL; # do not specify DELETE ... INPUT
```

# Instance Recovery in Real Application Clusters

An **instance** failure occurs when software or hardware problems disable an instance. After instance failure, Oracle automatically uses the online redo log file to perform database recovery as described in the following sections:

- Single Node Failure
- Multiple-Node Failures in Real Application Clusters
- Control File and Datafile Access for Instance Recovery in Real Application Clusters
- Steps of Oracle Instance Recovery in Real Application Clusters

**See Also:** *Oracle9i Backup and Recovery Concepts* for a general explanation of instance failure and recovery

## Single Node Failure

Instances in Real Application Clusters perform recovery through the SMON processes of the surviving instances. Instance recovery does not include restarting the failed instance or the recovery of applications that were running on the failed instance. Applications that were running can continue by using **failover** as described in *Oracle9i Real Application Clusters Setup and Configuration*.

When one instance performs recovery for another instance, the surviving instance reads redo log entries 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 recovery. Therefore, Oracle Corporation recommends that you mirror your online logs.

**See Also:** *Oracle9i Real Application Clusters Concepts* for conceptual information about application failover and **high availability**

## Multiple-Node Failures in Real Application Clusters

After multiple node failures, as long as one instance survives, its SMON process performs instance recovery for any other instances that fail. If all instances of a Real Application Clusters database fail, then Oracle performs failure recovery automatically the next time an instance opens the database.

The instance performing recovery does not have to be one of the instances that failed. In addition, the instance performing recovery can mount the database in either shared or exclusive mode from any node of a Real Application Clusters 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.

## Control File and Datafile Access for Instance Recovery in Real Application Clusters

If you use a recovery catalog, then RMAN uses it to recover the control file of the failed instance. If you do not use a recovery catalog, then RMAN uses a copy of the control file on the recovering instance to recover the control file of the failed instance.

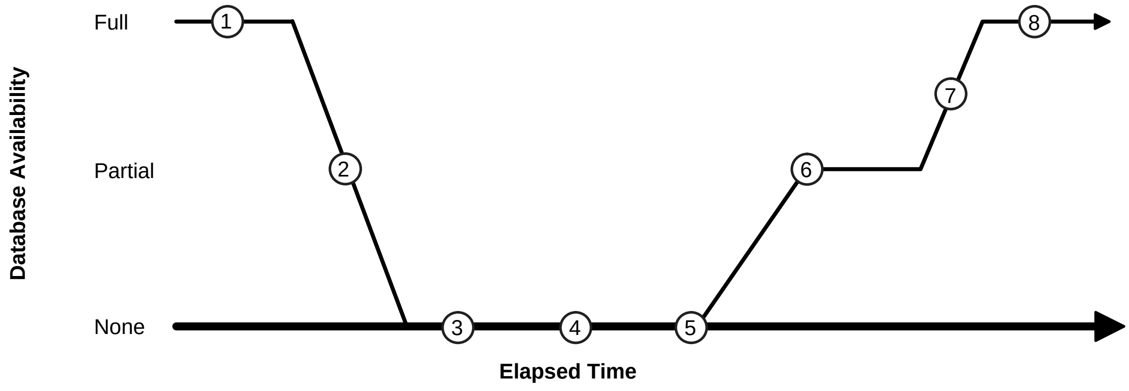
An instance performing recovery for a failed instance must also access all online datafiles that the failed instance accessed. When instance recovery fails because a datafile fails verification, Oracle writes a message to the alert log. After you correct the problem that prevented access to the file, use the SQL statement `ALTER SYSTEM CHECK DATAFILES` to verify the datafiles and make them available to the instance.

> **See Also:** *Oracle9i SQL Reference* for more information about the `CHECK DATAFILES` clause of the `ALTER SYSTEM` statement

## Steps of Oracle Instance Recovery in Real Application Clusters

Figure 7–1 and the narrated steps following illustrate the degree of database availability during each step of Oracle instance recovery.

***Figure 7–1 Steps of Oracle Instance Recovery***



The steps in recovery are:

1. Real Application Clusters is running on multiple nodes.
2. Node failure is detected.
3. The **Global Cache Service (GCS)** is reconfigured; resource management is redistributed onto the surviving nodes.
4. SMON reads the redo log of the failed instance to identify the database blocks that it needs to recover.
5. SMON issues requests to obtain all of the database blocks it needs for recovery. After the requests are complete, all other blocks are accessible.

> **Note:** The **Global Cache Service Processes (LMSn)** only re-master resources that lose their masters.

6. Oracle performs roll forward recovery: redo logs of the failed threads are applied to the database.
7. Oracle performs roll back recovery: undo blocks are applied to the database for all uncommitted transactions.
8. Instance recovery is complete and all data is accessible.

# Overview of Media Recovery in Real Application Clusters

Media failures occur when Oracle file storage media are damaged. Typically, a media failure prevents Oracle from reading or writing data, resulting in the loss of one or more database files. Media recovery must be user-initiated through a client application, whereas instance recovery is automatically performed by the database.

In these situations, use **Recovery Manager (RMAN)** to restore backups of the datafiles and then recover the database. The procedures for RMAN media recovery in Real Application Clusters environments do not differ substantially from the media recovery procedures for single-instance environments.

The issues for media recovery are the same as the issues described under the heading "Accessibility of Files and Backup Media" on page 7-4. 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 logs on disk or be able to restore them from backups.

# RMAN Restore Schemes for Real Application Clusters

This section describes the following restore schemes:

- Cluster File System Restore Scheme
- Non-CFS Restore Scheme

## Cluster File System Restore Scheme

If you made backups in a CFS scheme, then the restore and recovery procedures are simple and do not differ substantially from a single-instance scenario.

### Restore and Recovery of Nondistributed Backups in a CFS Scheme

Assume that you use the nondistributed backup scheme described under the heading "Backing Up to One Local Drive in the CFS Archiving Scheme" on page 7-7. This example requires the following channel configuration:

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

Assume that node 3 performs the backups. If node 3 is available for the restore and recovery processing, and *if* all the existing 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 tapes from node 3 available to this device.

### Restore and Recovery of Distributed Backups in a CFS Scheme

Assume that you use the distributed backup scheme described under the heading "Backing Up to Multiple Drives in the CFS Archiving Scheme" on page 7-7. Perform the following one-time configuration so that one channel is configured 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';
```

If all existing 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-CFS 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-CFS Archiving Scheme

Assume that you use the distributed backup scheme described under the heading "Backing Up to Multiple Drives in a Non-CFS Backup Scheme" on page 7-8. If all nodes are available and if all archived logs have been backed up, then you can perform a complete restore and recovery by mounting the database and executing 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 logs located on the local and remote disks. No manual transfer of logs is required.

### Restore and Recovery of Nondistributed Backups in a Non-CFS Archiving Scheme

Assume that you use the nondistributed backup scheme described under the heading "Backing Up to One Local Drive in a Non-CFS Archiving Scheme" on page 7-9. You have the following channel configuration:

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

Assume that node 3 performs the backups. If node 3 is available for the restore and recovery operation, and if the NFS mount points for the remote nodes are accessible, then run the following commands for complete recovery:

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

Note that if some of the nodes are down and you are prevented from accessing their logs through NFS, and if you do not have backups of the logs required for a complete recovery, then must perform an incomplete recovery up to the point of the first missing log after the whole database backup as in the following example:

```
RUN
{
  # in this example, sequence 1234 is the first missing log
  SET UNTIL LOG SEQUENCE 1234 THREAD 3;
  RESTORE DATABASE;
  RECOVER DATABASE;
}

ALTER DATABASE OPEN RESETLOGS;
```

# Parallel Recovery in Real Application Clusters

Parallel recovery uses multiple CPUs and I/O parallelism to reduce the time required to perform thread or media recovery. Parallel recovery is most effective at reducing recovery time while concurrently recovering several datafiles on several disks. You can use parallel instance recovery, parallel failure recovery, and parallel media recovery in Real Application Clusters databases.

> **See Also:** *Oracle9i User-Managed Backup and Recovery Guide* for more information on these topics

## Parallel Recovery with RMAN

With RMAN's RESTORE and RECOVER commands, Oracle automatically parallelizes the following three stages of recovery as described in this section:

- Restoring Datafiles
- Applying Incremental Backups
- Applying Redo Logs

### Restoring Datafiles

When restoring datafiles, the number of channels you allocate in the RMAN recover script effectively sets the parallelism 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 Redo Logs

RMAN applies redo logs using a specific number of parallel processes as determined by the setting for the RECOVERY_PARALLELISM initialization parameter.

> **See Also:** "Setting the RECOVERY_ PARALLELISM Parameter" on page 7-17

## Parallel Recovery with SQL*Plus

If you employ user-managed methods to back up and recover your database, then you can parallelize instance and media recovery using either of the following procedures:

- Setting the RECOVERY_ PARALLELISM Parameter
- Specifying RECOVER Statement Options

Real Application Clusters can use one process to read the log files sequentially and dispatch redo information to several recovery processes to apply the changes from the log files to the datafiles. Oracle automatically starts the recovery processes, so you do not need to use more than one session to perform recovery.

### Setting the RECOVERY_ PARALLELISM Parameter

The RECOVERY_PARALLELISM initialization parameter specifies the number of redo application server processes that participate in instance or media recovery. One process reads the log files sequentially and dispatches redo information to several recovery processes. The recovery processes then apply the changes from the 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.

### 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 Real Application Clusters database or the number of processes to distribute across all instances.

> **See Also:** Oracle system-specific documentation for more information on the allocation of recovery processes to instances

# Part IV

# Scalability in Real Application Clusters Environments

Part IV explains how to add nodes and instances and how to delete instances. The chapter in Part IV is:

# 8

# Adding Nodes and Instances and Deleting Instances in Real Application Clusters

This chapter provides detailed procedures on how to add nodes and instances and how to delete instances in **Real Application Clusters** database environments. The topics in this chapter are:

- Adding Nodes to a Cluster
- Overview of Procedures for Adding Nodes
- Adding a Node at the Clusterware Layer
- Adding a Node at the Oracle Layer
- Deleting Instances

> **See Also:** *Oracle9i Real Application Clusters Setup and Configuration* for procedures on using the **Database Configuration Assistant (DBCA)** to create and delete databases

# Adding Nodes to a Cluster

This section explains how to dynamically add nodes to Real Application Clusters environments. The topics in this section are:

- Overview of Procedures for Adding Nodes
- Adding a Node at the Clusterware Layer
- Adding a Node at the Oracle Layer

> **See Also:** "Deleting Instances" on page 8-21 for procedures on deleting an instance

# Overview of Procedures for Adding Nodes

The procedure for adding nodes to Real Application Clusters comprises two main steps:

- Adding a Node at the Clusterware Layer
- Adding a Node at the Oracle Layer

The method for adding nodes at the clusterware layer is operating-system dependent as described in this chapter. For the second step, adding a node at the Oracle layer, the procedures in this chapter require using the **Oracle Universal Installer (OUI)** and the **Database Configuration Assistant (DBCA)**. The procedures for the second step are generic and therefore apply to all platforms.

# Adding a Node at the Clusterware Layer

On UNIX, unless you preconfigure your clusterware to accommodate additional nodes, you cannot dynamically add nodes. Therefore, you must stop the clusterware and reconfigure your clusterware to accommodate additional nodes. Then follow the procedures under the heading "Adding a Node at the Clusterware Layer on UNIX".

> **Note:** You can dynamically add nodes on some platforms without preconfiguring your clusterware.

On Windows NT and Windows 2000, Oracle supplies the cluster software or **operating system-dependent (OSD) clusterware**. This enables you to dynamically add nodes as described under the heading "Adding a Node at the Clusterware Layer on Windows NT and Windows 2000" on page 8-4.

The topics in this section are:

- Adding a Node at the Clusterware Layer on UNIX
- Adding a Node at the Clusterware Layer on Windows NT and Windows 2000

## Adding a Node at the Clusterware Layer on UNIX

You can take advantage of the **scalability** and **high availability** features of Real Application Clusters by adding new nodes to existing UNIX clusters that uses a Real Application Clusters database. As mentioned, you can add nodes without stopping your database if you have preconfigured your UNIX cluster to accommodate additional nodes.

Oracle Corporation recommends using the following procedures to add new nodes to existing Real Application Clusters in UNIX environments. These procedures assume that there is already a UNIX cluster running your Real Application Clusters database.

To prepare the new node, ensure that it is an exact clone of the existing nodes in terms of vendor operating system, clusterware software, and so on. Then perform the following tasks in the order shown to add a node at the clusterware layer on UNIX:

1. Connect Node to Cluster
2. Install Vendor Cluster Software

3. Create Raw Devices (Optional)

### Connect Node to Cluster

Connect the new node’s hardware to the existing cluster. This includes electrical connections, network interconnects, shared disk subsystem connections, and so on. Refer to your hardware vendor documentation for details on this step.

### Install Vendor Cluster Software

Install the cluster software by referring to your vendor’s installation procedures. Also use your vendor documentation for procedures to make the new node an active member of your cluster.

### Create Raw Devices (Optional)

If your platform supports a cluster file system and you not are using raw devices for your cluster database, then you do not need to create raw devices as described in this section. Proceed to "Adding a Node at the Oracle Layer" on page 8-9.

As part of the preinstallation steps before installing the Real Application Clusters software on the new node, create at least two new disk volumes to accommodate the redo logs for the new instance. Refer to *Oracle9i Real Application Clusters Setup and Configuration* for the requirements for these redo logs and the preinstallation steps.

Make the disk volumes for the redo logs the same size as the redo log volumes that you configured for your existing nodes’ instances. Refer to your vendor documentation for procedures on creating disk volumes and raw devices.

If the cluster database to which you want to add a new node and instance uses automatic undo management, then create an additional raw volume for the new undo tablespace for the new instance. After completing this step, proceed to "Adding a Node at the Oracle Layer" on page 8-9 to add the node at the Oracle layer.

> **See Also:** *Oracle9i Real Application Clusters Setup and Configuration* for more information about creating raw devices

## Adding a Node at the Clusterware Layer on Windows NT and Windows 2000

You can take advantage of the scalability and high availability features of Real Application Clusters by dynamically adding new nodes to an existing Windows cluster that uses a Real Application Clusters database.

Oracle Corporation recommends using the following procedures to add a node in Windows environments. These procedures assume that there is already a Windows cluster running your Real Application Clusters database and that the new node uses the same operating system that is running on the existing nodes of your cluster. For example, you should have installed Microsoft Windows NT 4.0 or Windows 2000 on all the nodes in your cluster.

Perform the following tasks in the order shown to add a new node at the clusterware layer on Windows NT and Windows 2000:

1. Connect Node to Cluster
2. Install Cluster Software
3. Create Disk Partitions
4. Create Links to Disk Partitions

### Connect Node to Cluster

Connect the new node's hardware to the existing cluster. This includes electrical connections, network interconnects, shared disk subsystem connections, and so on. Refer to your hardware vendor documentation for details on this step.

### Install Cluster Software

If you are using your operating system vendor supplied clusterware on the existing nodes, also install this vendor software on the new node using your vendor's installation procedures.

However, if you are using Oracle Corporation's reference clusterware / operating system-dependent software on the existing nodes of your cluster, you must also install these Oracle OSD clusterware on the new node using the Cluster Setup Wizard. To do this, perform the following steps from one of the *existing* nodes before running the wizard:

1. If you use the **Virtual Interface Architecture (VIA)** interconnect on the existing cluster, make sure that VIA interconnect is also installed on the new node.
2. If you have a private interconnect network on the existing cluster, make sure that the private interconnect network is also installed on the new node.
3. If you are using a cluster file system, then skip this step: Determine the disk drive on which the cluster software was installed on the existing nodes. Make sure that you have at least 2MB of free space on the same drive on the new node to install the OSD clusterware and the Object Link Manager.

4. Make sure that you can execute the following from each of the existing nodes of your cluster:

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

Where the *host_name* is the public network name of the new node. You have administrative privileges on each node if the operating system responds with:

```
Command completed successfully.
```

5. To install the Oracle OSD clusterware when the new node already has a version of your vendor OSD clusterware installed, make sure the vendor OSD clusterware is stopped. Otherwise, continue to the next step.
6. Insert CD number one from the Oracle9*i* release 2 (9.2) CD set into the new node's CD-ROM drive.
7. Navigate to the `PREINSTALL\CLUSTERSETUP` directory, for example:

```
cd \PREINSTALL\CLUSTERSETUP
```

8. Run the Cluster Setup Wizard by entering the following from the `\PREINSTALL\CLUSTERSETUP` directory:

```
clustersetup
```

9. The cluster setup wizard should display its Welcome page. Click **Next** to proceed.
10. Check the **Node Addition** option and click **Next**.
11. Provide a public name for the new node you want to add. If the existing cluster also uses a high speed private network, provide a name for the private network interface for the new node and click **Next**.
12. Click **Finish**.

**See Also:**

- Cluster Setup Wizard online help
- *Oracle9i Database Installation Guide for Windows* for more information

### Create Disk Partitions

As part of the preinstall requirements before installing the Oracle database software on a node for Oracle Real Application Clusters, create raw disk partitions. Create at least two new disk partitions to accommodate the redo logs for the new instance. Make these disk partitions the same size as the redo log partitions that you configured for the existing nodes' instances. If the database to which you want to add a node uses automatic undo management, then create an additional logical partition for the undo tablespace.

> **See Also:** *Oracle9i Real Application Clusters Setup and Configuration* for more information on configuring raw partitions on Windows NT and Windows 2000 platforms

From one of the existing nodes of the cluster:

1. On Windows NT, start Disk Administrator using the path: Start > Programs > Oracle - [HOME_NAME] > Administrative Tools> Disk Administrator

   On Windows 2000, start Disk Management using the path: Start > Programs > Oracle - [HOME_NAME] > Administrative Tools > Computer Management. Then expand the Storage folder to Disk Management. For Windows 2000 only, use a *basic disk* as an extended partition for creating raw partitions.
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 the last four steps, 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. For Windows NT only, you may need to restart all the nodes in your cluster if you are using a pre-4.0 Service Pack. This makes all of the nodes recognize the new partitions.

> **Note:** The Pre-5 Service Packs are not Y2K compliant. Oracle Corporation therefore recommends using 6A.

### Create Links to Disk Partitions

You do not need to perform these procedures if you have a cluster file system.

Create the links to the partitions that you created in the "Create Disk Partitions" section so the existing nodes recognize these links. In addition, the new node should also recognize the pre-existing symbolic links to logical drives as well as the new links. To create these links, perform the following steps from one of the existing nodes:

1. Start the Object Link Manager by typing the following from the `%ORACLE_HOME%\bin` directory:

```
GUIOracleOBJManager
```

The Object Link Manager automatically detects the symbolic links to the logical drives and displays them in OLM's graphical user interface (GUI).

2. Recall the disk and partition numbers for the partitions that you created in the previous section "Create Disk Partitions". Look for the disk and partition numbers in the OLM GUI and perform the following tasks:

   a. Right-click next to the box under the **New Link** column and enter the link name for the first partition.

   b. Repeat step a. for the second and any additional partitions.

For example, if your Real Application Clusters 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, your link names for your redo logs are `db_redo3_1`, `db_redo3_2`, and so on.

3. If your existing database uses automatic undo management, then enter the link name for the logical partition for the undo tablespace that you created in the previous section "Create Disk Partitions". For example, if your Real Application Clusters 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 name for the undo tablespace should be *db*`_undotbs3`.

4. Select **Commit** from the Options menu. This creates the new links on the current node.

5. Select **Sync Nodes** from the Options menu. This makes the new links visible to the existing nodes of the cluster.

6. Select **Exit** from the Options menu to exit the Object Link Manager.

## Adding a Node at the Oracle Layer

At this point, you have added the new node to the cluster at the clusterware layer. To add a node to an existing cluster at the Oracle layer, on all platforms perform the following steps:

- Install the Oracle Software on the New Node
- Post-Installation Steps
- Add a Database Instance on the New Node

The following procedures explain these steps in detail.

**Install the Oracle Software on the New Node** Follow these steps to install the Oracle software on the *new node*:

1. If your platform supports a cluster file system then proceed to the next section titled "Post-Installation Steps" on page 8-10.
2. On the new node, insert the Oracle9*i* release 2 (9.2) CD-ROM into the new node’s CD-ROM drive.
3. Run the installer using one of the following procedures for your platform.
    - On UNIX, run the Installer on by executing the `runInstaller` command from the root directory of the first Oracle CD. The Installer displays the Welcome page.
    - On Windows NT and Windows 2000, run the Installer by executing the `setup.exe` command. The installer displays the Welcome page.
4. On the File Locations page in the Path field under the Destination heading, enter the `ORACLE_HOME` into which you are going to install the Oracle Enterprise Edition and Real Application Clusters software. The `ORACLE_HOME` you enter should be the same as the `ORACLE_HOME` that the existing nodes use.
5. On the Available Products page, select **Oracle9*i* Database**.
6. On the Install Types page, select **Enterprise Edition**. The Installer displays the Database Configuration page.
7. Follow the OUI instructions to install the same products and options that you installed on the existing nodes. Or select the **Software Only** configuration option on the Database Configuration page.
8. On the Node Selection page, select only the current or new node name. Do not select other pre-existing node names; the Oracle software is already on them.

9. When the OUI completes the installation of the Oracle software, run the post installation script `root.sh`. Then exit the Oracle Universal Installer and proceed to the next section, "Post-Installation Steps".

**Post-Installation Steps** From the *new node*, perform the following post-installation steps:

1. Run the Oracle Net Configuration Assistant from the `bin` directory in `ORACLE_HOME` to configure a new **listener** for the node.
2. Complete any post-installation steps for your platform by referring to your platform-specific Oracle documentation.

**Add a Database Instance on the New Node** To add an instance on the new node, make sure that the **Global Services Daemon (GSD)** is running on all the existing nodes as well as on the new node. If the GSD is not running, then from any of the nodes execute the `gsdctl start` command from the following platform-specific locations:

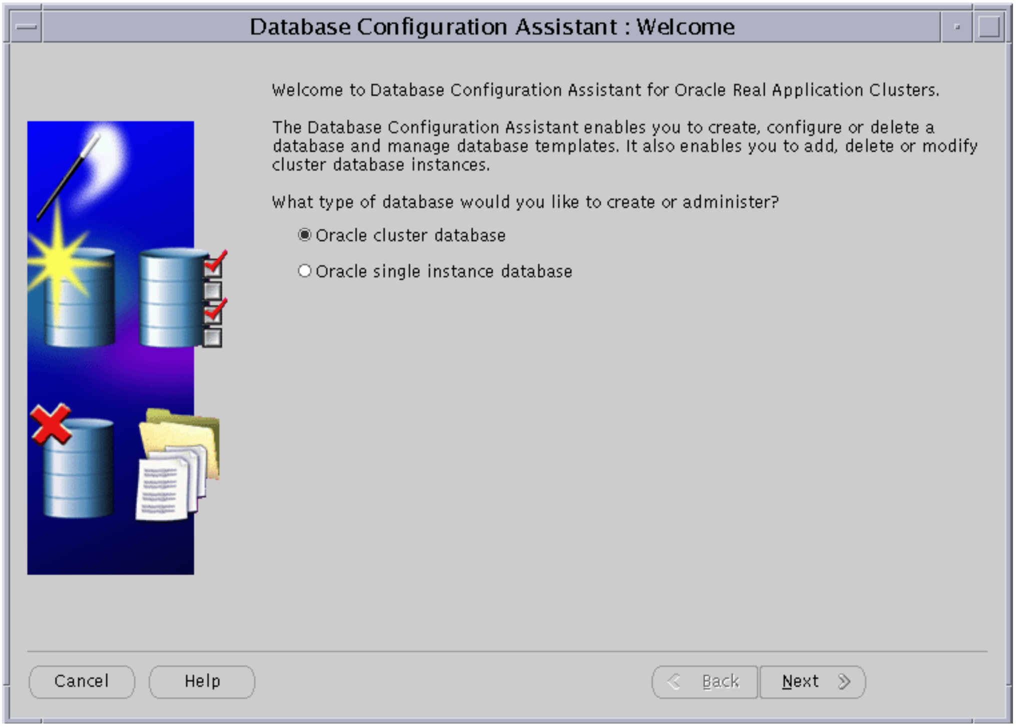
- On UNIX from `$ORACLE_HOME/bin`
- On Windows NT and Windows 2000 from `%ORACLE_HOME%\bin`

Then go to one of the existing nodes and perform the following steps:

1. Start the Database Configuration Assistant by typing `dbca` in the `bin` directory in `$ORACLE_HOME` on UNIX, or on Windows NT and Windows 2000, choose Start > Programs > Oracle - HOME_NAME > Configuration and Migration Tools > Database Configuration Assistant

The DBCA wizard starts and displays the Welcome page for Real Application Clusters shown in Figure 8–1.

***Figure 8–1 DBCA Welcome Page for Real Application Clusters***



Database Configuration Assistant : Welcome

Welcome to Database Configuration Assistant for Oracle Real Application Clusters.

The Database Configuration Assistant enables you to create, configure or delete a database and manage database templates. It also enables you to add, delete or modify cluster database instances.

What type of database would you like to create or administer?

● Oracle cluster database

○ Oracle single instance database

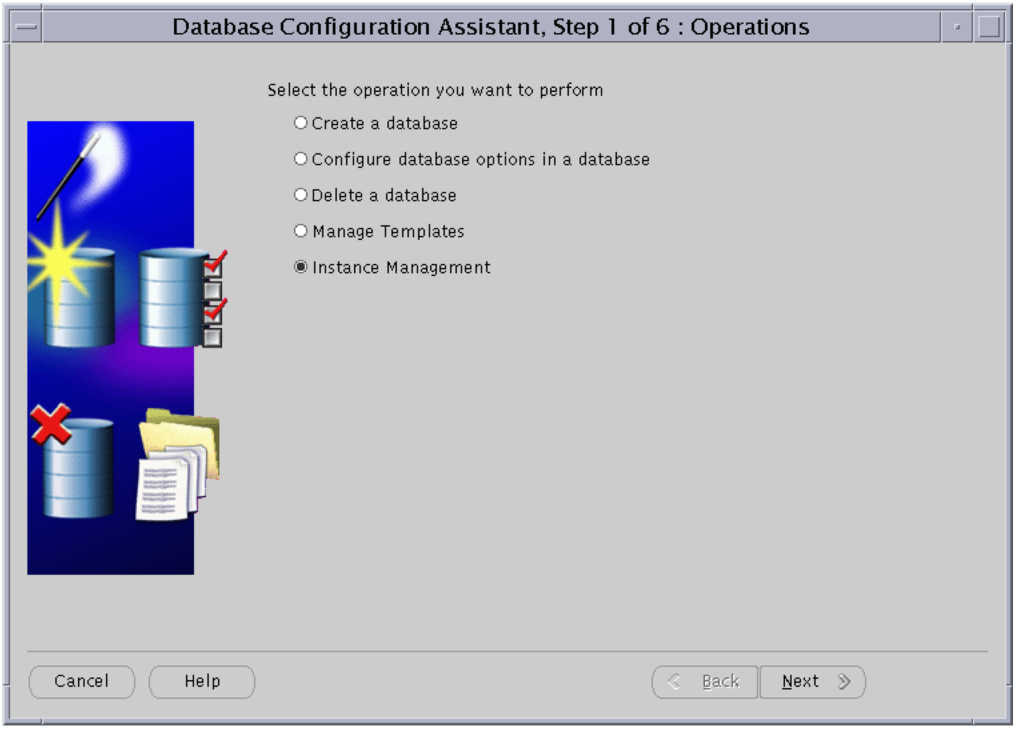
**Note:** If the Welcome page for Real Application Clusters does not display, then the DBCA was unable to:

- Communicate with the **Cluster Manager (CM)** software on Windows NT or Windows 2000
- Detect the **Global Cache Service (GCS)** software or the lists of nodes in the cluster on UNIX operating systems

To resolve this, refer to your vendor’s operating system-dependent documentation to correct the problem and restart the DBCA.

2. Select **Oracle cluster database** and click **Next**. After you click **Next**, the DBCA displays the Operations page as shown in Figure 8–2:

***Figure 8–2 Database Configuration Assistant Operations Page***



Database Configuration Assistant, Step 1 of 6 : Operations

Select the operation you want to perform

○ Create a database

○ Configure database options in a database

○ Delete a database

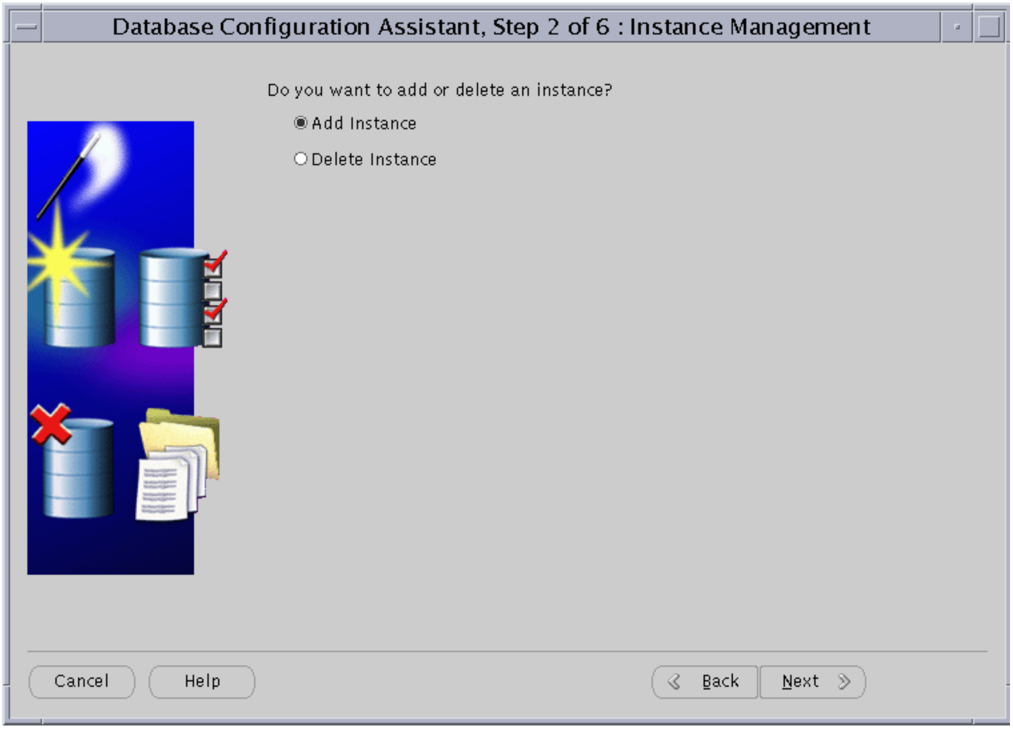
○ Manage Templates

● Instance Management

3. Select **Instance Management** and click **Next**.

After you click **Next**, the DBCA displays the Instance Management page shown in Figure 8–3.

*Figure 8–3 Database Configuration Assistant Instance Management Page*



Database Configuration Assistant, Step 2 of 6 : Instance Management

Do you want to add or delete an instance?

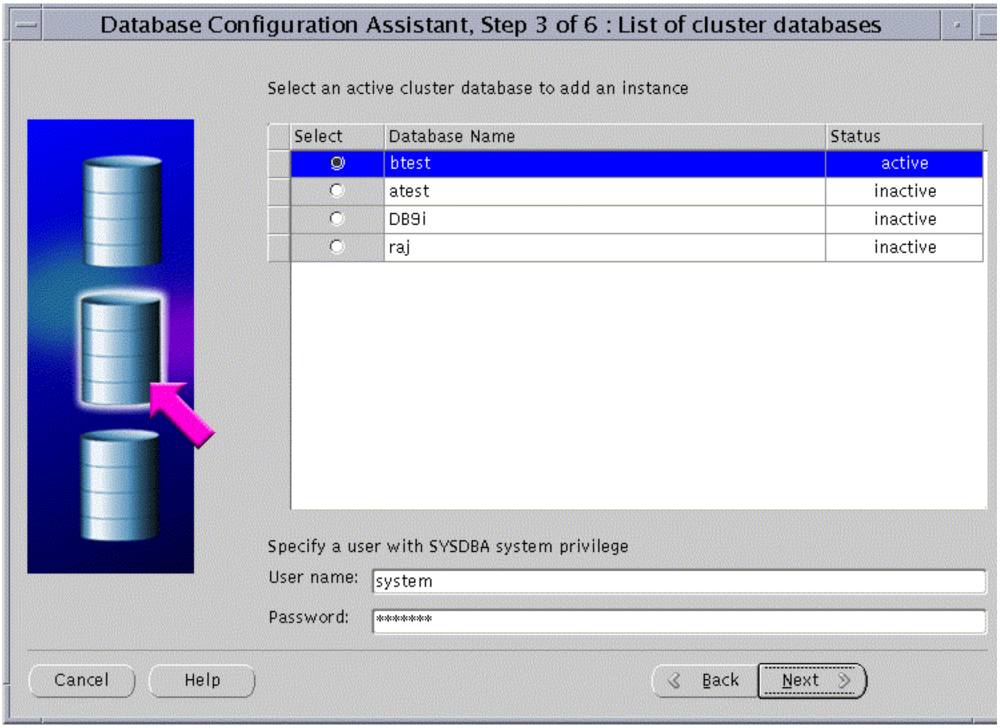
● Add Instance

○ Delete Instance

4. Select **Add Instance** and click **Next**.

After you click **Next**, the DBCA displays the List of Cluster Databases page that shows the databases and their current statuses, such as `ACTIVE`, or `INACTIVE`, as shown in Figure 8–4.

***Figure 8–4 Database Configuration Assistant List of Cluster Databases Page***



Database Configuration Assistant, Step 3 of 6 : List of cluster databases

Select an active cluster database to add an instance

| Select | Database Name | Status |
|---|---|---|
| ● | btest | active |
| ○ | atest | inactive |
| ○ | DB9i | inactive |
| ○ | raj | inactive |

Specify a user with SYSDBA system privilege

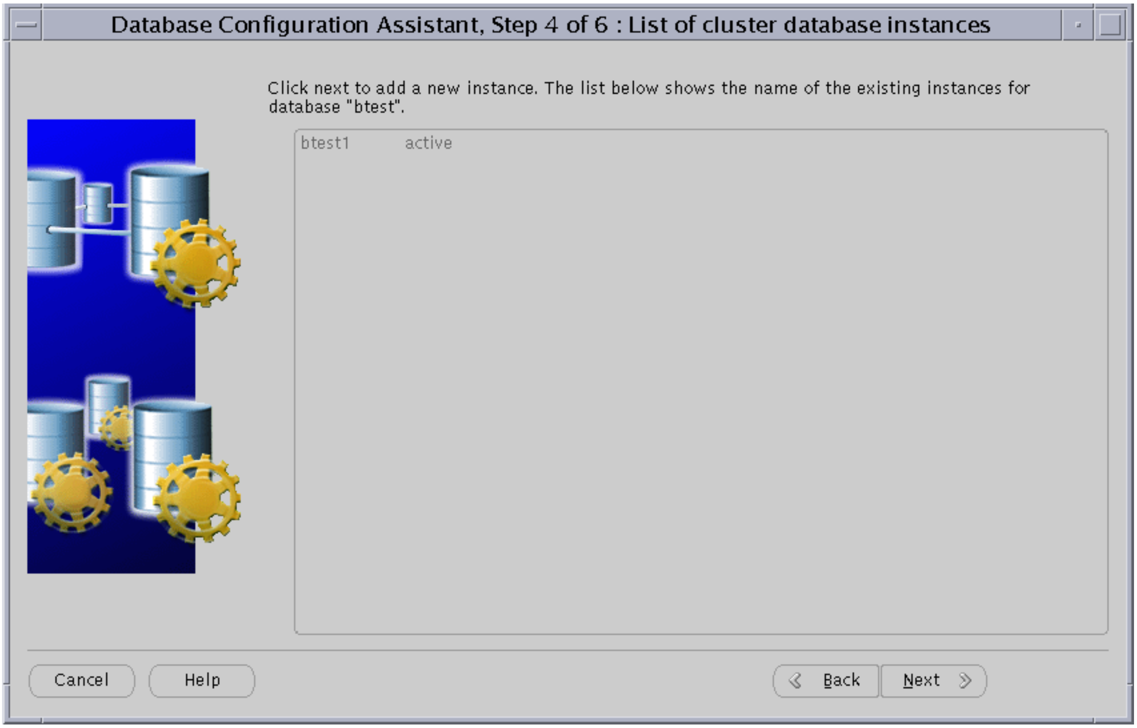
User name: system

Password: ********

5. Select an active Real Application Clusters database to which you want to add an instance from the List of Cluster Databases page. 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, then enter a valid user ID and password and click **Next**.

After you click **Next**, the DBCA displays the List of Cluster Database Instances page showing the names of the instances for the Real Application Clusters database that you selected as shown in Figure 8–5.

***Figure 8–5 Database Configuration Assistant List of Cluster Database Instances Page***



Database Configuration Assistant, Step 4 of 6 : List of cluster database instances

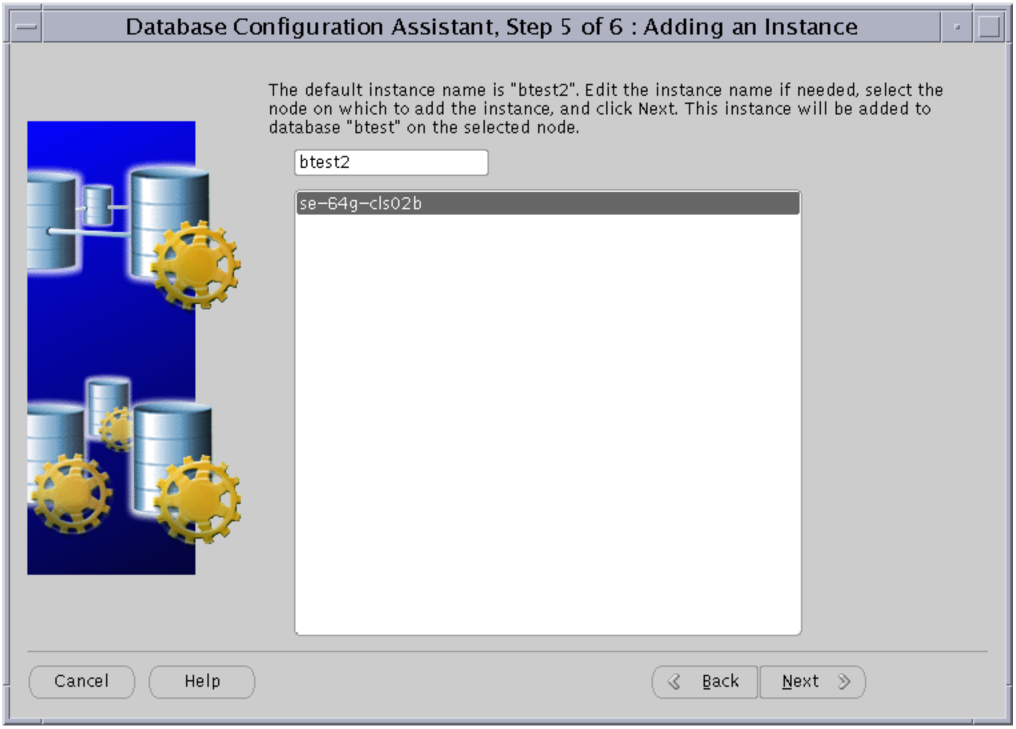
Click next to add a new instance. The list below shows the name of the existing instances for database "btest".

btest1 active

6. Click **Next**.

After you click **Next**, the DBCA displays the Adding an Instance page shown in Figure 8–6.

***Figure 8–6 Database Configuration Assistant Adding an Instance Page***



Database Configuration Assistant, Step 5 of 6 : Adding an Instance

The default instance name is "btest2". Edit the instance name if needed, select the node on which to add the instance, and click Next. This instance will be added to database "btest" on the selected node.

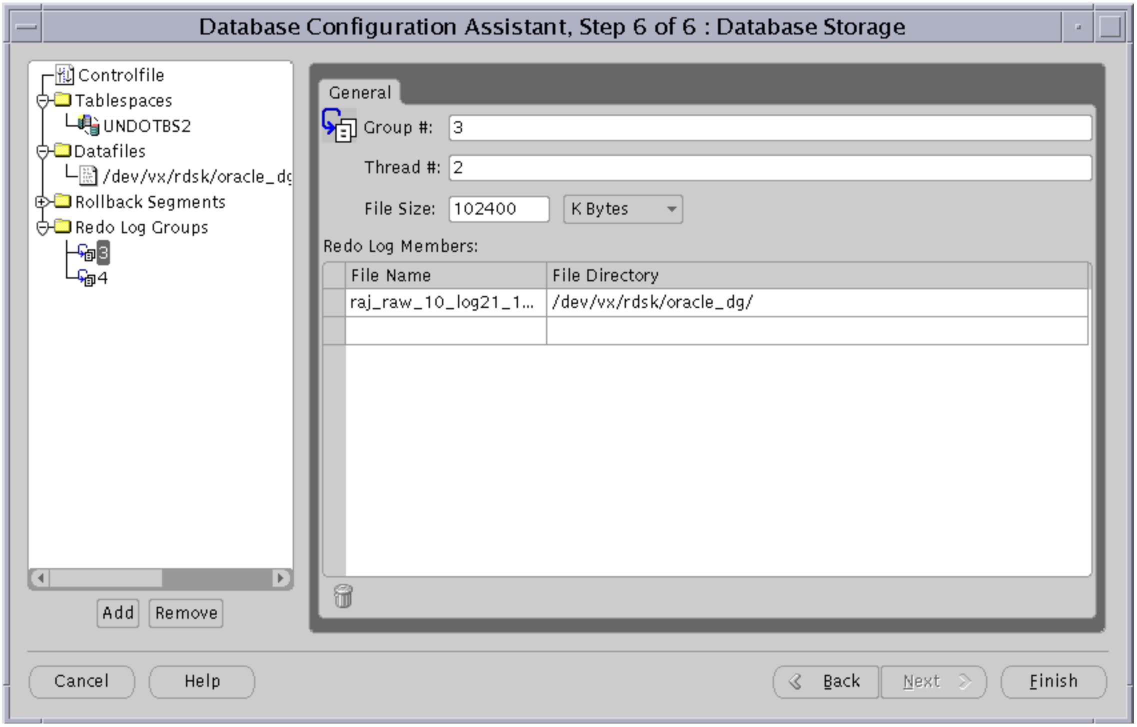
btest2

se-64g-cls02b

7. 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 and click **Next**.

After you click **Next**, the DBCA displays the Database Storage page as shown in Figure 8–7.

***Figure 8–7 Database Configuration Assistant Database Storage Page***



Database Configuration Assistant, Step 6 of 6 : Database Storage

- Controlfile
- Tablespaces
  - UNDOTBS2
- Datafiles
  - /dev/vx/rdsk/oracle_dg
- Rollback Segments
- Redo Log Groups
  - 3
  - 4

General

Group #: 3

Thread #: 2

File Size: 102400 K Bytes

Redo Log Members:

| File Name | File Directory |
|---|---|
| raj_raw_10_log21_1... | /dev/vx/rdsk/oracle_dg/ |

8. If your database uses automatic undo management and your platform supports a cluster file system, then skip this step unless you want to change the default undo tablespace datafile name that the DBCA provides. Otherwise, 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. You can also change the name to the cluster file system file name if you do not want to use the default name. Oracle Corporation does *not* recommend that you change the storage properties for the undo tablespace.

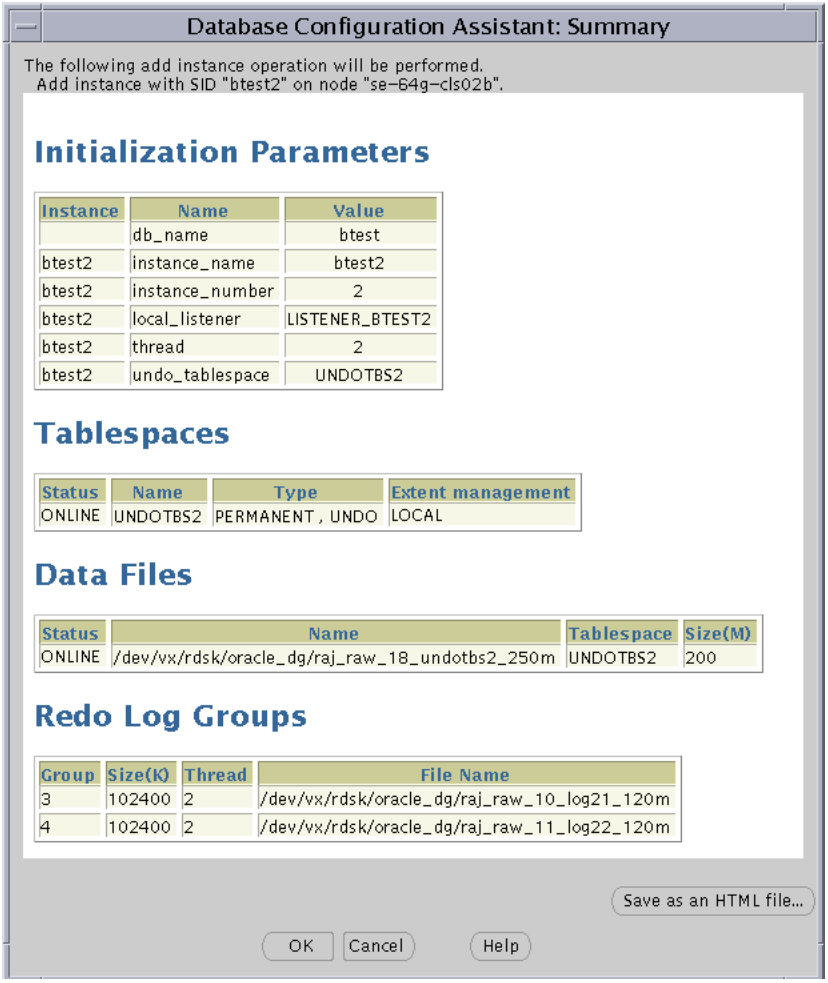
9. If your database uses rollback undo management, then select the **Rollback Segment** folder to change the rollback segment names if you do not want to use the default rollback segment names that the DBCA provides. Oracle

Corporation does *not* recommend that you change the storage properties for the rollback segments.

10. If your platform supports a cluster file system, then skip this step unless you want to change the default redo log group file names. Otherwise, select the **Redo Log Groups** folder and expand it. For each redo log group number you select, the DBCA displays another dialog box. For UNIX platforms, enter the raw device name that you created in the section "Create Raw Devices (Optional)" on page 8-4 in the **File Name** field. On Windows NT or Windows 2000 platforms, enter the symbolic link name that you created in the section, "Create Links to Disk Partitions" on page 8-8.
11. Repeat the previous step for all other redo log groups and click **Finish**.

After you click **Finish**, the DBCA displays a Summary dialog similar to the dialog in Figure 8–8.

***Figure 8–8 Database Configuration Assistant Summary Dialog***



The following add instance operation will be performed.
Add instance with SID "btest2" on node "se-64g-cls02b".

Initialization Parameters

| Instance | Name | Value |
|---|---|---|
| | db_name | btest |
| btest2 | instance_name | btest2 |
| btest2 | instance_number | 2 |
| btest2 | local_listener | LISTENER_BTEST2 |
| btest2 | thread | 2 |
| btest2 | undo_tablespace | UNDOTBS2 |

Tablespaces

| Status | Name | Type | Extent management |
|---|---|---|---|
| ONLINE | UNDOTBS2 | PERMANENT , UNDO | LOCAL |

Data Files

| Status | Name | Tablespace | Size(M) |
|---|---|---|---|
| ONLINE | /dev/vx/rdsk/oracle_dg/raj_raw_18_undotbs2_250m | UNDOTBS2 | 200 |

Redo Log Groups

| Group | Size(K) | Thread | File Name |
|---|---|---|---|
| 3 | 102400 | 2 | /dev/vx/rdsk/oracle_dg/raj_raw_10_log21_120m |
| 4 | 102400 | 2 | /dev/vx/rdsk/oracle_dg/raj_raw_11_log22_120m |

**12.** Review the information on the dialog and click **OK**, or click **Cancel** to end the instance addition operation.

If you click **OK**, then the DBCA displays a progress dialog that shows the DBCA performing the instance addition operation. During this operation the DBCA adds the instance and creates the instance's Oracle Net configuration. When the DBCA completes the instance addition operation, the DBCA displays a dialog asking whether you want to perform another operation. Click **No** to exit the DBCA, or click **Yes** and the DBCA displays the Operations screen shown in Figure 8–3 on page 8-13.

If you click **No** and continue the instance addition operation, then at this point you have accomplished the following:

- Created a new database instance on the new node
- Created the new instance's required services for Windows NT or Windows 2000
- Configured Oracle Net
- Started the required services for Windows NT or Windows 2000
- Started the new instance and its listeners

# Deleting Instances

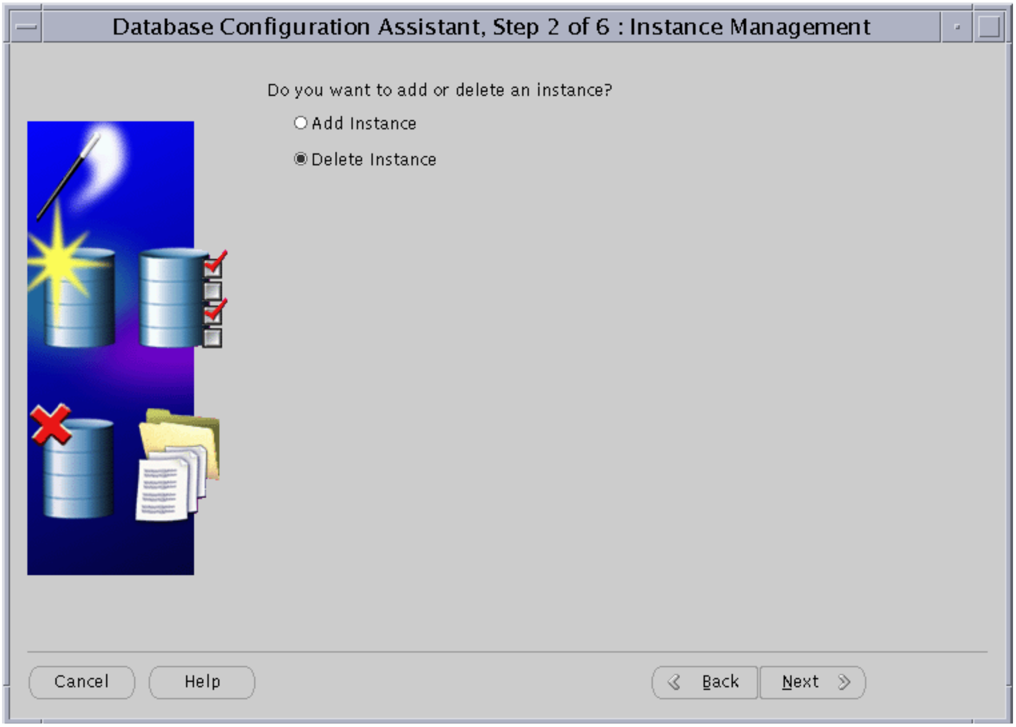
The following procedures explain how to delete an instance with the Database Configuration Assistant. Where noted, refer to previous figures in this chapter. To delete an instance:

1. Go to the DBCA Operations Page shown in Figure 8–2, select **Instance Management** and click **Next**.

After you click **Next**, the DBCA displays the Instance Management page shown in Figure 8–9.

2. Select **Delete Instance** and click **Next**.

***Figure 8–9 DBCA Instance Management Page***



Database Configuration Assistant, Step 2 of 6 : Instance Management

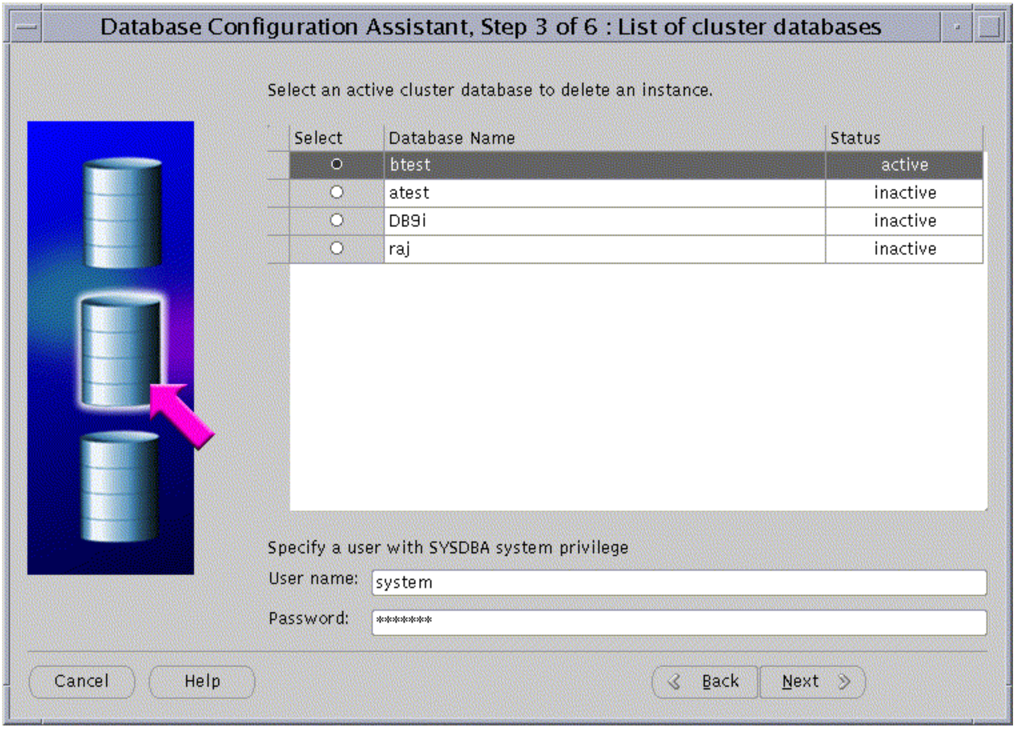
Do you want to add or delete an instance?

○ Add Instance

● Delete Instance

After you click **Next**, the DBCA displays the List of Cluster Databases page shown in Figure 8–10.

***Figure 8–10 Database Configuration Assistant List of Cluster Databases***



Database Configuration Assistant, Step 3 of 6 : List of cluster databases

Select an active cluster database to delete an instance.

| Select | Database Name | Status |
|---|---|---|
| ○ | btest | active |
| ○ | atest | inactive |
| ○ | DB9i | inactive |
| ○ | raj | inactive |

Specify a user with SYSDBA system privilege

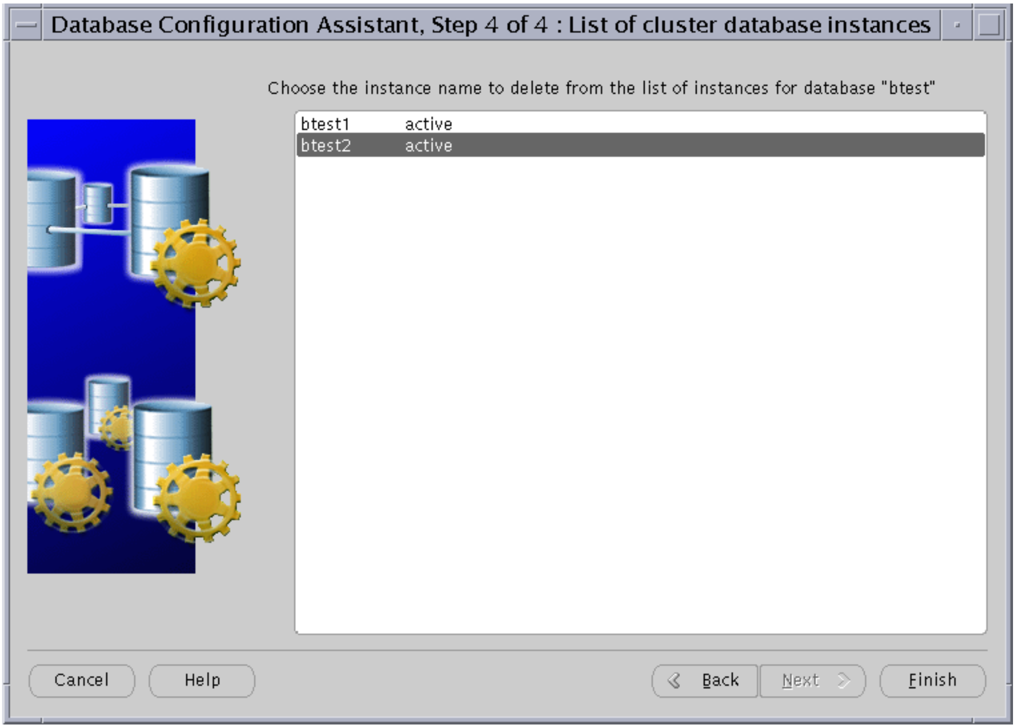
User name: system

Password: ********

3. Select a Real Application Clusters 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**.

After you click **Next**, the DBCA displays the List of Cluster Database Instances page shown in Figure 8–11. The List of Cluster Database Instances page shows the instances associated with the Real Application Clusters database that you selected and the status of each instance.

***Figure 8–11 Database Configuration Assistant List of Cluster Database Instances***



Database Configuration Assistant, Step 4 of 4 : List of cluster database instances

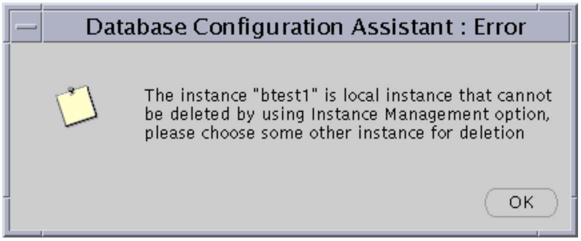
Choose the instance name to delete from the list of instances for database "btest"

```
btest1      active
btest2      active
```

4. Select the instance to delete and click **Finish**.

If you select the local instance, then the DBCA displays an Error dialog as shown in Figure 8–12.

***Figure 8–12 Database Configuration Assistant Error Dialog***



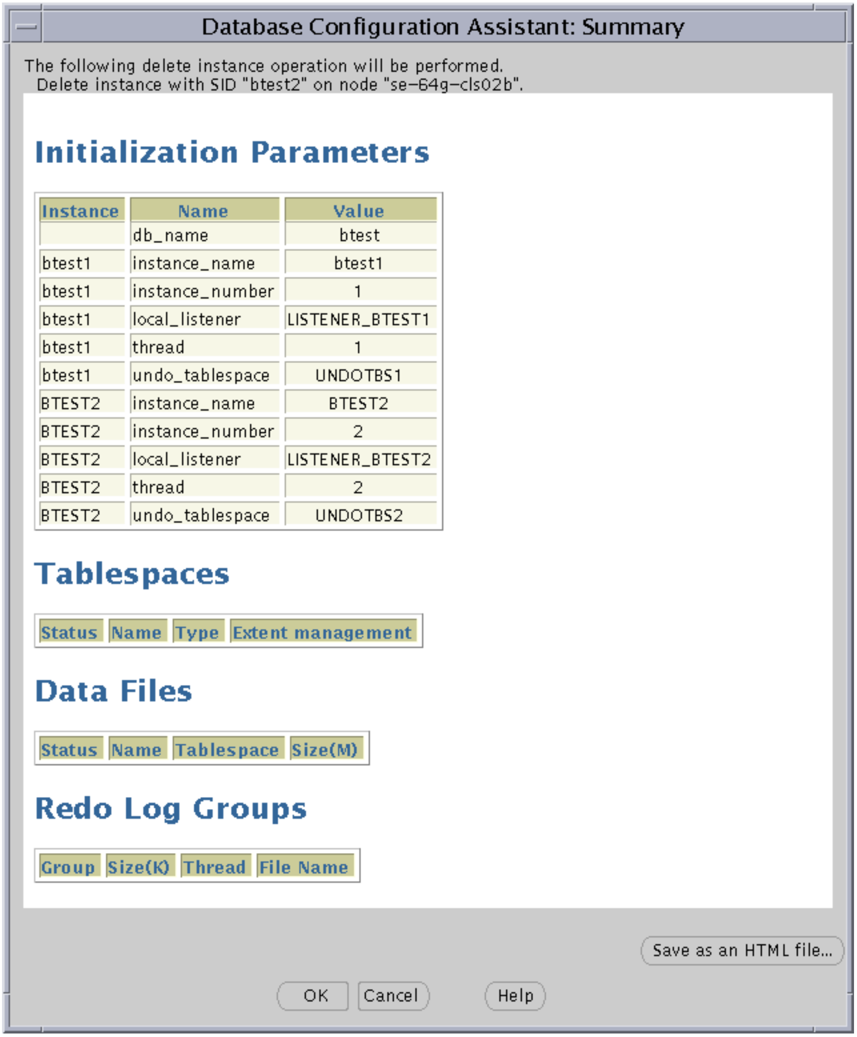
Database Configuration Assistant : Error

The instance "btest1" is local instance that cannot be deleted by using Instance Management option, please choose some other instance for deletion

5. If the DBCA displays this error dialog, then click **OK** and reselect another instance on the List of Cluster Database Instances page and click **Finish**.

6. The DBCA displays the DBCA Summary dialog similar to the dialog in Figure 8–13.

***Figure 8–13 Database Configuration Assistant Summary Dialog***



Database Configuration Assistant: Summary

The following delete instance operation will be performed.
Delete instance with SID "btest2" on node "se-64g-cls02b".

## Initialization Parameters

| Instance | Name | Value |
|---|---|---|
| | db_name | btest |
| btest1 | instance_name | btest1 |
| btest1 | instance_number | 1 |
| btest1 | local_listener | LISTENER_BTEST1 |
| btest1 | thread | 1 |
| btest1 | undo_tablespace | UNDOTBS1 |
| BTEST2 | instance_name | BTEST2 |
| BTEST2 | instance_number | 2 |
| BTEST2 | local_listener | LISTENER_BTEST2 |
| BTEST2 | thread | 2 |
| BTEST2 | undo_tablespace | UNDOTBS2 |

## Tablespaces

| Status | Name | Type | Extent management |
|---|---|---|---|

## Data Files

| Status | Name | Tablespace | Size(M) |
|---|---|---|---|

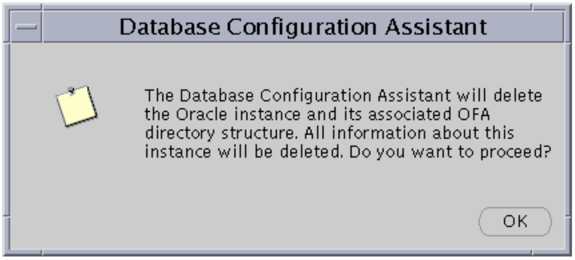
## Redo Log Groups

| Group | Size(K) | Thread | File Name |
|---|---|---|---|

7. Review the information about the instance deletion operation on the Summary dialog and click **OK**. Otherwise, click **Cancel** to end the instance deletion operation.

If you click **OK**, then the DBCA displays a Confirmation dialog similar to the dialog in Figure 8–14.

***Figure 8–14 Database Configuration Assistant Confirmation Dialog***



Database Configuration Assistant

The Database Configuration Assistant will delete the Oracle instance and its associated OFA directory structure. All information about this instance will be deleted. Do you want to proceed?

8. Click **OK** on the Confirmation dialog to proceed with the instance deletion operation.

After you click **OK**, 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. Click **No** and exit the DBCA or click **Yes** to perform another operation. If you click **Yes**, then the DBCA displays the Operations page shown in Figure 8–3 on page 8-13.

At this point, you have accomplished the following:

- Stopped the listeners associated with the selected instance
- Deleted the selected database instance from the instance's configured node
- Deleted the selected instance's services for Windows NT or Windows 2000
- Removed the Oracle Net configuration
- Deleted the Oracle Flexible Architecture directory structure from the instance's configured node

# Part V

## Reference

Part V provides reference information for Real Application Clusters. The contents of Part V are:

# A

# Troubleshooting

This appendix explains how to manage trace files and how to contact Oracle Support Services. The topics in this appendix are:

- Using Trace Files
- Contacting Oracle Support Services

> **See Also:** *Oracle9i Real Application Clusters Setup and Configuration* for information on troubleshooting **Oracle Enterprise Manager** Service Discovery failures.

# Using Trace Files

This section discusses the following trace file subjects:

- Background Thread Trace Files
- User Thread Trace Files
- Alert File
- Error Call Trace Stack

## Background Thread Trace Files

**Real Application Clusters** background threads use trace files to record occurrences and exceptions of database operations as well as errors. These detailed trace logs are helpful to Oracle support to debug problems in your **cluster database** configuration. Background thread trace files are created regardless of whether the `BACKGROUND_DUMP_DEST` parameter is set in the **initdbname.ora initialization parameter file**. If you set `BACKGROUND_DUMP_DEST`, the trace files are stored in the directory specified. If you do not set the parameter, the trace files are stored in:

- `$ORACLE_HOME/admin/`*db_name*`/bdump` on UNIX operating systems
- `%ORACLE_HOME%\admin\`*`db_name`*`\bdump` on Windows NT and Windows 2000 operating systems

The Oracle database creates a different trace file for each background thread. The name of the trace file contains the name of the background thread followed by the extension `.trc`, such as:

- *sid*`dbwr.trc`
- *sid*`smon.trc`

Trace information is reported in the trace files shown in Table A–1:

***Table A–1 Background Thread Trace Files***

| Trace File | Description |
|---|---|
| *sid*`lckn.trc` | Trace file for the **Global Cache Service Processes (LMSn)**. This trace file shows lock requests for other background processes. |
| *sid*`lmdn.trc` | Trace file for the LMD*n* process. This trace file shows lock requests. |

| Trace File | Description |
|---|---|
| *sid*lmon.trc | Trace file for the LMON process. This trace file shows the status of the cluster. |
| *sid*p00n.trc | Trace file for the parallel execution processes. |

## User Thread Trace Files

Trace files are also created for user threads if the USER_DUMP_DEST parameter is set in the initialization parameter file. The trace files for the user threads have the form ora*xxxxx*.trc, where *xxxxx* is a 5-digit number indicating the process identifier (PID) on UNIX or the **thread number** on Windows NT and Windows 2000.

## Alert File

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

The *sid*alrt.log file is in the directory specified by the BACKGROUND_DUMP_DEST parameter in the init*db_name*.ora initialization parameter file. If you do not set the BACKGROUND_DUMP_DEST parameter, the *sid*alrt.log file is generated in:

- $ORACLE_BASE/admin/*db_name*/bdump on UNIX operating systems
- %ORACLE_HOME%\admin\*db_name*\bdump on Windows NT and Windows 2000 operating systems

## Error Call Trace Stack

Oracle Worldwide Support might ask you to create an error call trace stack for a particular trace file. An error call trace stack provides a program trace of specific background or user threads in the database.

To create an error call trace stack:

1. Obtain the Oracle process ID for the background processes:

```
SELECT pid "Oracle Process Id",
       name
   from V$PROCESS, V$BGPROCESS
   where V$PROCESS.addr = V$BGPROCESS.paddr;
```

The output displayed should resemble the following:

```
Oracle Pro NAME
---------- -----
         2 PMON
         3 LMON
         4 DBW0
         5 LGWR
         6 CKPT
         7 SMON
         8 RECO
         9 SNP0
        10 SNP1
        11 GCS0
```

2. Dump the trace stack to the trace file. For example, to dump the trace stack of LMON, enter:

   a. Set the Oracle process ID to LMON. The process ID is 3 in this example:

      ```
      ORADEBUG setorapid 3
      ```

   b. Dump the error stack to *sid*`lmon.trc`:

      ```
      ORADEBUG dump errorstack 3
      ```

## Contacting Oracle Support Services

If after reading this appendix, you still cannot resolve your problems, submit a TAR by way of Metalink to report the error to Oracle Support Services. Please include the following information:

- Your cluster hardware specifications, for example, a two-**node** cluster of Dell PowerEdge 6100 servers
- Your operating system version
- All five digits of the release number of your Oracle RDBMS (for example, 9.2.0.0.1)
- All five digits in the release number of your Oracle9*i* Real Application Clusters database
- On Windows NT and Windows 2000, the contents of `HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\OSD` key

- Cluster OSD upgrades from vendor
- Information about the particular operation that failed, for example, database startup or query
- A list of steps that reproduce the problem

## Severe Errors

If an `ORA-600` error appears, it will appear in the `*sid*alrt.log` file. If an ORA-600 error or any other severe errors appear in the `*sid*alrt.log` file, then provide all files in:

- `$ORACLE_HOME/admin/db_name/bdump` on UNIX operating systems
- `%ORACLE_HOME%\admin\db_name\bdump` on Windows NT and Windows 2000 operating systems

# B

# Associating Instances and Users with Free Lists and Free List Groups (Optional)

This appendix explains how to associate instances and users with free lists and free list groups. It also discusses SQL-specific free list options and the preallocation of extents to free list groups. Only use the information in this appendix if you cannot use automatic segment-space management. The topics in this appendix are:

- Associating Instances with Free Lists
- Associating User Processes with Free Lists and Free List Groups
- SQL Options for Managing Free Space with Free Lists
- Preallocating Extents to Free List Groups

> **Note:** Oracle Corporation strongly recommends that you use automatic segment-space management instead of free lists and free list groups.

# Associating Instances with Free Lists

In general, all tables should have the same number of **free list groups**, but the number of free lists within a group may vary, depending on the type and amount of activity of each table. The **Cache Fusion** resources that often cover blocks in one free list group tend to be held primarily by the instance using that free list group. This is because an instance that modifies data is usually more likely to reuse that data than other instances. If multiple instances take free space from the same extent, then they are more likely to contend for blocks in that extent if they subsequently modify the data that they inserted. Automatic segment-space management, however, eliminates this performance issue in Real Application Clusters.

## Assignment of New Instances to Existing Free List Groups

If `MAXINSTANCES` is greater than the number of free list groups in the table or cluster, then an **instance number** maps to the free list group associated with:

*instance_number* modulo *number_of_free_list_groups*

*Modulo* (or *rem* for *remainder*) is a formula for determining which free list group should be used by calculating a remainder value. In the following example there are 2 free list groups and 10 instances. To determine which free list group instance 6 will use, the formula would read 6 modulo 2 = 0. Six divided by 2 is 3 with zero remainder, so instance 6 will use free list group 0. Similarly, instance 5 would use free list group 1 because 5 modulo 2 = 1. Five is divisible by 2 with a remainder of 1.

If there are more free list groups than `MAXINSTANCES`, then a different hashing mechanism is used. If multiple instances share one free list group, they share access to every extent specifically allocated to any instance sharing that free list group.

## FREELIST GROUPS and MAXINSTANCES

In a system with relatively few **nodes**, the `FREELIST GROUPS` option for a table should generally have the same value as the `MAXINSTANCES` option of `CREATE DATABASE`, which limits the number of instances that can access a database concurrently. In a Massively Parallel Processing system, however, `MAXINSTANCES` could be many times larger than `FREELIST GROUPS` so that many instances share one group of free lists.

**See Also:** *Oracle9i Real Application Clusters Deployment and Performance* for more information on associating instances and users with free list groups

## Associating User Processes with Free Lists and Free List Groups

User processes associate with free lists based on their Oracle process IDs. Each user process has access to only one free list in the free list group for the instance on which it is running. Every user process also has access to the **master free list** of free blocks. If a table has multiple free lists but does not have multiple free list groups, or if it has fewer free list groups than the number of instances, then free lists and free list groups would be shared among user processes from different instances.

Using the statement ALTER SESSION INSTANCE_NUMBER, you can increase the instance number value beyond the value of MAXINSTANCES. You can also dynamically alter a table's free list assignment with the ALTER TABLE statement. However, you cannot use this statement for free list groups.

## SQL Options for Managing Free Space with Free Lists

Several SQL options enable you to create free lists and free list groups for tables, clusters, and indexes. You can explicitly specify that new space for an object be taken from a specific datafile. You can also associate free space with particular free list groups that you can then associate with particular instances.

The SQL statements include:

```
CREATE [TABLE | CLUSTER | INDEX]
  STORAGE
  FREELISTS
  FREELIST GROUPS
ALTER [TABLE | CLUSTER | INDEX]
  ALLOCATE EXTENT
  SIZE
  DATAFILE
  INSTANCE
```

You can use these SQL options with the INSTANCE_NUMBER parameter to associate data blocks with instances.

**See Also:** *Oracle9i SQL Reference* for complete syntax of these statements

## Preallocating Extents to Free List Groups

Preallocating extents is a static approach to the problem of preventing automatic allocation of extents by Oracle. You can preallocate extents to tables that have free list groups. This means that all free blocks are formatted into free lists that reside in the free list group of the instance to which you are preallocating the extent. This approach is useful if you need to accommodate objects that you expect will grow in size.

# C

# Real Application Clusters Management Tools Error Messages

This appendix describes the Real Application Clusters management tools error messages. The groups of error messages in this appendix appear alphabetically. The error message groups in this appendix are:

- Global Services Daemon Error Messages (PRKA)
- Cluster Command Error Messages (PRKC)
- Cluster Setup for Windows NT Error Messages (PRKI)
- Server Control (SRVCTL) Utility Error Messages (PRKO)
- Cluster Database Management Error Messages (PRKP)
- Repository Error Messages (PRKR)

> **See Also:** *Oracle9i Database Administrator's Guide for Windows* for Windows-based error messages and *Oracle9i Database Error Messages* for all other error messages

# Global Services Daemon Error Messages (PRKA)

**PRKA-2001: GSD already exists**

**Cause:** An attempt was made to create the Global Services Daemon (GSD) application while there is a GSD application already running.

**Action:** You cannot create the GSD application at this moment since it already exists. Stop the running GSD and create the GSD application again using 'srvctl add nodeapps'. You can also use the existing GSD application.

**PRKA-2002: Listener already exists**

**Cause:** An attempt was made to create a Listener application while there is a running Listener application.

**Action:** You cannot create the Listener application at this moment since it already exists. Stop the running Listener and create the Listener application again using 'srvctl add nodeapps'. You can also use the existing Listener application.

**PRKA-2003: Agent already exists**

**Cause:** An attempt was made to create an Agent application while there is a running Agent application.

**Action:** You cannot create the Agent application at this moment since it already exists. Stop the running Agent and create the Agent application again using 'srvctl add nodeapps'. You can also use the existing Agent application.

**PRKA-2004: VIP already exists**

**Cause:** An attempt was made to create a Virtual IP application while there is a running Virtual IP application.

**Action:** You cannot create the Virtual IP application at this moment since it already exists. Stop the running Virtual IP application and create the Virtual IP application using 'srvctl add nodeapps'. You can also use the existing Virtual IP application.

**PRKA-2006: Netmasks are different**

**Cause:** Invalid range of VIP addresses specified for the cluster while doing 'srvctl add vip_range'

**Action:** Check if the starting and ending addresses in the VIP range being added, belong to the same netmask.

### PRKA-2007: No free VIP is available

**Cause:** No free Virtual IP address available from the list of Virtual IP addresses added to the cluster database. All of the of Virtual IP addresses added are in use.

**Action:** Add new Virtual IP addresses using to the cluster database using 'srvctl add vip_range'

### PRKA-2008: VIP is not found

**Cause:** A Virtual IP address was not found in the Virtual IP range added to the cluster database.

**Action:** Add Virtual IP addresses to the cluster using 'srvctl add vip_range'

### PRKA-2009: Remote command failed

**Cause:** An error occurred while trying to execute an operation by the Global Services Daemon either on a local node or a remote node.

**Action:** Check whether the Global Services Daemon is up and responsive by doing 'srvctl config'.

### PRKA-2010: Failed to get the VIP Range configuration in formation from the OCR

**Cause:** An error occurred while trying to get the Virtual IP range configuration from the Configuration Repository. Either the Global Services Daemon is not up or the Configuration Repository is not accessible. The configuration repository may not be accessible if the repository is corrupted or for insufficient privileges.

**Action:** Check if the Global Services Daemon is up and the configuration repository is accessible by doing 'srvctl config'

### PRKA-2011: Failed to add the VIP Range configuration in formation to the OCR

**Cause:** An error occurred while trying to add a Virtual IP range configuration to the Configuration Repository. Either the Global Services Daemon is not up or the Configuration Repository is not accessible. The configuration repository may not be accessible if the repository is corrupted or for insufficient privileges.

**Action:** Check if the Global Services Daemon is up and the configuration repository is accessible by doing 'srvctl config'

# Cluster Command Error Messages (PRKC)

**PRKC-1000 "No active nodes detected in the cluster"**

**Cause:** The active node list of the cluster could not be retrieved. This could occur if the operating system dependent 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 'lsnodes' binary from ORACLE_HOME/bin and see if it prints the node list correctly. Use vendor documentation to get additional information about the current state of the operating system dependent clusterware.

**PRKC-1001 "Error submitting commands in the buffer"**

**Cause:** An internal error has occurred.

**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:** Check that all nodes are up and the communication between the nodes is working.

**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. This can occur either because one or more nodes failed during the operation or the destination directory did not have write permission for the user on one or more of the nodes or the source file did not exist or on UNIX platforms 'rcp' command to one or more nodes failed.

**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 has permission to do 'rcp' 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. This can occur either because one or more nodes failed during the operation or the destination directory did not have write permission for the user on one or more of the nodes or the source file did not exist or on UNIX platforms ’rcp’ command to one or more nodes failed.

**Action:** See earlier error messages 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. This can occur either because one or more nodes failed during the operation or the destination directory did not have write permission for the user on one or more of the nodes or the source file did not exist or on UNIX platforms ’rcp’ command to one or more nodes failed.

**Action:** See earlier error messages 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. This can occur either because one or more nodes failed during the operation or the destination directory did not have write permission for the user on one or more of the nodes or the source file did not exist or on UNIX platforms ’rcp’ command to one or more nodes failed.

**Action:** See earlier error messages 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:** See earlier error messages.

### 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:** See earlier error messages.

### 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:** See earlier error messages.

### 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 operating system dependent clusterware was not functioning properly or there is already a running GSD.

**Action:** Check the state of the clusterware by running ’lsnodes’ from ORACLE_HOME/bin and following vendor documentation. 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 operating system dependent 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 operating system dependent clusterware cannot be contacted. This can occur because the operating system dependent clusterware was not functioning properly.

**Action:** Check the state of the operating system dependent clusterware using ’lsnodes’ and following vendor documentation.

**PRKC-1022 "Could not get "node name" for node {0} in {1}"**

**Cause:** The operating system dependent clusterware cannot be contacted. This can occur because the operating system dependent clusterware was not functioning properly.

**Action:** Check the state of the operating system dependent clusterware using ’lsnodes’ and following vendor documentation.

## Cluster Setup for Windows NT Error Messages (PRKI)

**PRKI-2059 "Unable to copy files to the nodes"**

**Cause:** Lack of permission to create destination file or copy file to the destination location.

**Action:** Verify permission to create the destination file. Also, please make sure that the destination file is not currently in use.

**PRKI-2119 "Unable to delete files on {0}. Install may not succeed if files are not deleted"**

**Cause:** Files we are trying to delete do not exist, or are currently being used.

**Action:** If the files we are trying to delete do not exist, ignore this error. If the files we are trying to delete are currently being used, please end the processes that are using the files and then retry.

**PRKI-2060 "Unable to create service on the nodes"**

**Cause:** Lack of permission to create the service, or service already exists.

**Action:** Verify permission to create service on destination node or nodes. Also, verify that the service is not already existing.

**PRKI-2061 "Unable to start service on the nodes"**

**Cause:** Service is not created, service is already running, service marked for deletion, service could not be started properly.

**Action:** Try to start the service manually from the service control panel and check the Windows Error message that it gives.

**PRKI-2064 "Unable to update registry entries on the nodes"**

**Cause:** Lack of permission to modify registry entries on the nodes, error while updating the registries.

**Action:** Verify permission to modify registry entries.

**PRKI-2066 "Unable to find an Oracle disk partition. Please exit from the wizard, create Oracle partitions and try again."**

**Cause:** Absence of Oracle disk partition, failure to load OLM dll.

**Action:** Verify that OLM dll (oobjlib.dll) is present in the load path. Also, verify that Oracle disk partitions are present by going to the Disk Administrator and looking for partitions.

**PRKI-2114 "Cannot collect and verify hardware information for all the nodes. Press abort to exit the wizard or ignore to continue."**

**Cause:** Failure to connect to the remote nodes while adding a node to a cluster. Failure to collect VIA information if VIA was chosen.

**Action:** If you are trying to add a node to an already existing cluster, try to map a drive letter to a drive on each of the remote node (using commands like "net use"). Also, verify the we have permission to start temporary service on remote nodes that will collect VIA information, in case VIA was chosen.

**PRKI-2116 "Cannot connect to remote drive on node {0}"**

**Cause:** Failure to connect to remote node.

**Action:** Try to map a driver letter on local node to the remote node using commands like "net use".

**PRKI-2120 "Unable to delete service {0} on {1}. Install may not succeed if services are not deleted"**

**Cause:** Service does not exist, or service could not be deleted.

**Action:** Check the service control panel to see if the service exists. If the service does not exist, ignore this error. If the service exists, try to delete it from the command line using utilities like "net" and "sc". If this fails, check the Windows error message returned by these commands.

## Server Control (SRVCTL) Utility Error Messages (PRKO)

**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 when getting Cluster Database configuration for the named database.

**Action:** Make sure that the database has been configured in Cluster Database Configuration Repository; make sure that GSDs are running on each node in the cluster.

**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 'lsnodes' 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>

**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; make sure that GSDs are running on each node in the cluster.

**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; make sure that GSDs are running on each node in the cluster.

**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; make sure that GSDs are running on each node in the cluster.

**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; make sure that GSDs are running on each node in the cluster.

**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; make sure that GSDs are running on each node in the cluster.

**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; make sure that GSDs are running on each node in the cluster.

**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 and if the database instance on the node is listed in SID_NAME in the listener configuration; make sure that GSDs are running on each node in the cluster.

## Cluster Database Management Error Messages (PRKP)

**PRKP-1000 "Cannot retrive 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 start" 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 start 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 all the listeners associated with instance {0} on node{1}"**

**Cause:** The listener name associated with an instance could not be determined.

**Action:** Ensure that listener.ora contains a SID_LIST entry for the named instance

**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-1040 "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

## Repository Error Messages (PRKR)

**PRKR-1001 "cluster database {0} does not exist"**

**Cause:** The cluster database was never registered in the repository.

**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 register a cluster database which was already existing in the repository.

**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 was never registered in the repository.

**Action:** Use ’srvctl config’ to discover all the databases and their instances.

**PRKR-1004 "instance {0} already exists"**

**Cause:** An attempt was made to register an instance which already existed.

**Action:** See earlier messages.

**PRKR-1005 "adding of cluster database {0} configuration failed, {1}"**

**Cause:** An error occurred while attempting to add a new cluster database in the configuration repository. Either the Global Services Daemons were not up on all the nodes in the cluster or the configuration repository was not accessible or the configuration repository was never initialized.

**Action:** Use ’gsdctl stat’ on each node to check if the Global Services Daemons are up. If the Global Services Daemon is not up on any node use ’gsdctl start’ to start it. Check if the configuration repository is accessible by using ’srvctl config’. Use ’srvconfig -init’ to initialize the configuration repository if it was never initialized.

**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:** Contact your customer support representative

**PRKR-1012 "error {0} while serializing configuration of cluster database {1}"**

**Cause:**

**Action:** Contact your customer support representative

**PRKR-1013 "error {0} while deserializing configuration of cluster database {1}"**

**Cause:**

**Action:** Contact your customer support representative

**PRKR-1014 "error {0} while serializing directory"**

**Cause:**

**Action:** Contact your customer support representative

**PRKR-1015 "error {0} while deserializing directory"**

**Cause:**

**Action:** Contact your customer support representative

**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:**

**Action:** Contact your customer support representative

**PRKR-1019 "writing of directory failed, {0}, {1}"**

**Cause:**

**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 configuration repository.

**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 management repository was never initialized.

**Action:** Use ’srvconfig -init’ to initialize the repository.

**PRKR-1038 "invalid argument {0} specified to -init option"**

**Cause:** Invalid argument specified to ’srvconfig -init’

**Action:** See usage of ’srvconfig -init’ for details.

**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 repository.

**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 configuration repository 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 management repository 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 d id 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:**

**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 an 8.1.7 or previous database in the management repository and the file argument passed was not of <cluster database name>.conf form.

**Action:** See usage of srvconfig for details.

### PRKR-1053 "invalid range {0} specified in node_list = {1}"

**Cause:** Detected an invalid parameter specified in <opsName>.conf file while attempting conversion of 8.l.7 or previous version of Oracle Parallel Server into RAC.

**Action:** Check 8.1.7 documentation and make sure that format of <opsName>.conf file is consistent with the documentation.

### PRKR-1054 "invalid parameter {0} specified in inst_oracle_sid = {1}"

**Cause:** Detected an invalid parameter specified in <opsName>.conf file while attempting to add a 8.l.7 or previous version of Oracle Parallel Server database in the configuration repository.

**Action:** Check 8.1.7 documentation and make sure that format of <opsName>.conf file is consistent with the documentation.

### PRKR-1055 "invalid extra arguments {0} specified to {1} option"

**Cause:** Extra number of arguments provided 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 previous version of Oracle Parallel Server in the configuration repository.

**Action:** See earlier error messages.

**PRKR-1057 "environment variable does not exist"**

**Cause:** Attempted to retrieve non existing environment variable.

**Action:** Set the environment variable.

# Glossary

### auto-discovery

When you execute the Discover Node command from the **Console**, the Management Server contacts the Oracle Intelligent Agent installed on that **node** to discover the Oracle services installed on the node. The Management Server then places the new information in the repository, and updates the hierarchical tree in the Navigator window of the Console, displaying a broad view of all nodes and their respective services.

### cache coherency

The synchronization of data in multiple caches so that reading a memory location by way of any cache will return the most recent data written to that location by way of any other cache. Sometimes called *cache consistency*.

### Cache Fusion

A diskless cache coherency mechanism in **Real Application Clusters** that provides copies of blocks directly from a holding instance's memory cache to a requesting instance's memory cache.

### cluster

A set of instances that cooperates to perform the same task.

### cluster database

The generic term a **Real Application Clusters** database.

### clustered database

See **cluster database**.

### clustering

See **cluster database**.

### Cluster Manager (CM)

An operating system-dependent component that discovers and tracks the membership state of each **node** by providing a common view of membership across the **cluster**. The CM also monitors process health, specifically the health of the database instance. The **Global Enqueue Service Monitor (LMON)**, a background process that monitors the health of the **Global Cache Service (GCS)**, registers and de-registers from the CM.

### CM

See: **Cluster Manager (CM)**.

### connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

### connect-time failover

See: **failover**.

### connection load balancing

A feature that balances the number of active connections among various instances and **shared server dispatchers** for the same service. Because of service registration's ability to register with remote listeners, a listener is always aware of all instances and dispatchers regardless. This way, a listener can sends an incoming client request for a specific service to the least loaded instance and least loaded dispatcher regardless of its location.

### Console

The **Oracle Enterprise Manager** Console gives you a central point of control for the Oracle environment through an intuitive graphical user interface (GUI) that provides powerful and robust system management. The Console provides menus, tool bars, and launch palettes that enable access to Oracle tools. The Console consists of four separate windows. See also: **Navigator Window**,**Group Window**, **Event Management Window**, and **job window**.

### daemon

Disk and Execution Monitor. A program that is not invoked explicitly, but lies dormant waiting for some condition to occur.

### Database Configuration Assistant (DBCA)

An Oracle tool for creating and deleting databases and for managing database templates.

### dedicated server

A server that requires a dedicated server process for each user process. There is one server process for each client. Oracle Net sends the address of an existing server process back to the client. The client then resends its connect request to the server address provided. Contrast with **shared server**.

### decision support system (DSS)

Database and application environments that help with decision support or data warehouse systems.

### dispatcher

A process that enables many clients to connect to the same server without the need for a **dedicated server** process for each client. A dispatcher handles and directs multiple incoming network session requests to shared server processes. See also **shared server**.

### distributed recovery

The process of recovery over a cluster.

### DSS

See: **decision support system (DSS)**.

### enqueues

Shared memory structures that serialize access to database resources and are associated with a session or transaction. In Real Application Clusters, enqueues can be global to a database. Enqueues are local to one instance if Real Application Clusters is not enabled.

### Enterprise Manager

See: **Oracle Enterprise Manager**.

### Enterprise Manager Configuration Assistant (EMCA)

A tool for creating, deleting, and modifying **Oracle Enterprise Manager** configurations and settings.

### Event Management Window

Part of the **Console**. The Event Management window enables the administrator to remotely monitor critical database and system events.

### exclusive (X) access mode

A write-only global block access mode. In this mode no other access is allowed.

### failover

The process of failure recognition and recovery.

### free list groups

Sets of free lists available for use by one or more instances.

### General Purpose

The Database Configuration Assistant's preconfigured database template for a hybrid database environment. This template includes datafiles.

### Global Cache Service (GCS)

Process that implement **Cache Fusion**. It maintains the block mode for blocks in the global role. It is responsible for block transfers between instances. The Global Cache Service employs various background processes such as the **Global Cache Service Processes (LMSn)** and **Global Enqueue Service Daemon (LMD)**.

### Global Cache Service Processes (LMS*n*)

Processes that manage remote **Global Cache Service (GCS)** messages. **Real Application Clusters** provides for up to 10 Global Cache Service Processes. The number of LMS*n* varies depending on the amount of messaging traffic among nodes in the cluster.

### Global Cache Service (GCS) resources

Global resources that coordinate access to data blocks in the buffer caches of multiple **Real Application Clusters** instances to provide cache coherency.

### global database name

The full name of the database that uniquely identifies it from any other database. The global database name is of the form `database_name.database_domain`—for example: `OP.US.ORACLE.COM`

### global dynamic performance views (GV$)

Dynamic performance views storing information about *all* open instances in a Real Application Clusters cluster. (Not only the local instance.) In contrast, standard dynamic performance views (V$) only store information about the local instance.

### Global Enqueue Service (GES)

A service that coordinates enqueues that are shared globally.

### Global Enqueue Service Daemon (LMD)

The resource agent process that manages requests for **Global Cache Service (GCS)** resources to control access to blocks. The LMD process also handles deadlock detection and remote resource requests. Remote resource requests are requests originating from another instance.

### Global Enqueue Service Monitor (LMON)

The background LMON process monitors the entire cluster to manage global resources. LMON manages instance *deaths* and the associated recovery for the **Global Cache Service (GCS)**. In particular, LMON handles the part of recovery associated with global resources. LMON-provided services are also known as Cluster Group Services.

### Globalization Support

An architecture that stores, processes, and retrieves data in native languages. It ensures that database utilities and error messages, sort order, date, time, monetary, numeric, and calendar conventions automatically adapt to the native language and locale.

### global resources

Inter-instance synchronization mechanisms that provide cache coherency for Real Application Clusters. The term can refer to both **Global Cache Service (GCS)** resources and **Global Enqueue Service (GES)** resources.

### Global Services Daemon (GSD)

A component that receives requests from SRVCTL to execute administrative job tasks, such as startup or shutdown. The command is executed locally on each **node**, and the results are returned to SRVCTL. GSD is installed on the nodes by default.

### Group Window

Part of the **Console**. The Group window provides a customized graphical representation of key objects. It is created by the administrator.

### GSD

See **Global Services Daemon (GSD)**.

### GV$

See: **global dynamic performance views (GV$)**.

### high availability

Systems with redundant components that provide consistent and uninterrupted service, even in the event of hardware or software failures. This involves some degree of redundancy.

### hybrid

A hybrid database is one that has both OLTP and Data Warehouse processing characteristics. See **General Purpose**.

### IFILE

A parameter designating the continuation file of an **initialization parameter file**.

### initialization parameter file

Files that contains information to initialize the database such as **spfile.ora**, **initdbname.ora** and **initsid.ora**.

### initsid.ora

An instance initialization parameter file that contains parameters unique for an instance and points to **initdbname.ora** for database parameters.

### initdbname.ora

A common database initialization parameter file shared among the instance that contains database parameters.

### instance

For a **Real Application Clusters** database, each **node** within a cluster usually has one instance of the running Oracle software that references the database. When a database is started, Oracle allocates a memory area called the System Global Area (SGA) and starts one or more Oracle processes. This combination of the SGA and the Oracle processes is called an instance. Each instance has unique Oracle System Identifier (*sid*), instance name, rollback segments, and thread ID.

### instance membership recovery (IMR)

The method used by **Real Application Clusters** guaranteeing that all cluster members are functional or active. IMR polls and arbitrates the membership. Any members that do not show a heartbeat by way of the control file or who do not respond to periodic activity inquiry messages are presumed terminated.

### instance name

Represents the name of the instance and is used to uniquely identify a specific instance when clusters share common services names. The instance name is identified by the `INSTANCE_NAME` parameter in the instance initialization file, initsid.ora. The instance name is the same as the Oracle System Identifier (*sid*.) See also: **Oracle system identifier (sid)**.

### instance number

A number that associates extents of data blocks with particular instances. The instance number enables you to start up an instance and ensure that it uses the extents allocated to it for inserts and updates. This will ensure that it does not use space allocated for other instances.

### Intelligent Agent

See: **Oracle Intelligent Agent**.

### interconnect

The communication link between nodes.

### Inter-Process Communication (IPC)

A high-speed operating system-dependent transport component. The IPC transfers messages between instances on different nodes. Also referred to as the **interconnect**.

### job window

Part of the **Console**. The Job window enables the administrator to automate repetitive activities.

### latch

A low-level serialization mechanism that protects in-memory data structures in the System Global Area (SGA). Latches do not protect datafiles, are automatic, and are held for a very short time in exclusive mode. Because latches are synchronized within a **node**, they do not facilitate internode synchronization.

### LDAP

See **Lightweight Directory Access Protocol (LDAP)**.

### Lightweight Directory Access Protocol (LDAP)

A protocol for accessing on-line directory services.

### listener

A process that resides on the server to listen for incoming client connection requests and manage the traffic to the server. When a client requests a network session with a server a listener receives the request. If the client information matches the listener information, then the listener grants a connection to the server.

### listener.ora

A listener configuration file that identifies the protocol addresses on which the listener is accepting connection requests and the services the listener listens for.

### LMD

See **Global Enqueue Service Daemon (LMD)**.

### LMON

See **Global Enqueue Service Monitor (LMON)**.

### LMS*n*

See **Global Cache Service Processes (LMSn)**.

### Management Server

The **Oracle Enterprise Manager** Management Server provides centralized intelligence and distributed control between the **Console** and the managed **nodes**, and processes system management tasks sent by the Console and administers the distribution of these tasks across the enterprise. The Management Server stores all system data, application data, and information about the state of managed nodes in a repository. The repository is a set of tables stored in a database. High performance and scalability is ensured because the workload is automatically shared and balanced when there are multiple Management Servers.

### master free list

A list of blocks containing available space drawn from any extent in a table.

### mean time between failures (MTBF)

The average time (usually expressed in hours) that a component works without failure. It is calculated by dividing the total number of failures into the total number of operating hours observed. The term can also mean the length of time a user can reasonably expect a device or system to work before an failure occurs.

### mean time to failure (MTTF)

The average period of time that a component will work until failure.

### mean time to recover (MTTR)

The average time that it takes to get a failed piece of hardware back on line. Outside the context of Real Application Clusters, the acronym MTTR is also used for Mean Time to Repair.

### MTBF

See: **mean time between failures (MTBF)**.

### MTTF

See: **mean time to failure (MTTF)**.

### MTTR

See: **mean time to recover (MTTR)**.

### multiple oracle homes

The capability of having more than one Oracle home directory on a machine.

### N

See: **null (N)**.

### naming method

The method used by a client application to resolve a **net service name** to a connect descriptor.

### Navigator Window

Part of the **Console**. The Navigator window contains an object browser that provides an organized, consistent, and hierarchical view of the database objects in the network.

### net service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a net service name in a connect string for the desired service.

### network file system (NFS)

A protocol developed by Sun Microsystems, and defined in RFC 1094, that enables a computer to access files over a network as if the files were on local disks.

### node

A node is a machine on which an instance resides.

### null (N)

Null indicates that an access mode is not assigned to a block or resource.

### OLTP

See: **online transaction processing (OLTP)**.

### online transaction processing (OLTP)

The processing of transactions by computers in real time.

### operating system-dependent (OSD) clusterware

Software that consists of several software components developed by Oracle or by other vendors. The OSD clusterware maps the key operating system and clusterware services required for proper operation of **Real Application Clusters**.

### Oracle Enterprise Edition

Oracle Enterprise Edition is an Object-Relational Database Management System (ORDBMS). It provides the applications and files to manage a database. All other Real Application Clusters components are layered on top of Oracle Enterprise Edition.

### Oracle Enterprise Manager

A system management tool that provides an integrated solution for centrally managing your heterogeneous environment. Oracle Enterprise Manager combines a graphical **Console**, **Management Server**, **Oracle Intelligent Agent**, **repository database**, and tools to provide an integrated, comprehensive systems management platform for managing Oracle products.

### Oracle Intelligent Agent

A process that runs on each of the nodes. It functions as the executor of jobs and events sent by the **Console** by way of the **Management Server**. High availability is ensured since the agent can function regardless of the status of the **Console** or network connections.

### Oracle Net

A software component that enables connectivity. It includes a core communication layer called the Oracle Net foundation layer and network protocol support. Oracle Net enables services and their applications to reside on different computers and communicate as peer applications.

### Oracle Performance Manager

An add-on application for **Oracle Enterprise Manager** that offers tabular and graphic performance statistics for **Real Application Clusters**. The statistics represent the aggregate performance for all instances running on Real Applications.

### Oracle Real Application Clusters

A breakthough architecture that enables clusters to access a shared database. Real Application Clusters includes the software component that provides the necessary Real Application Clusters scripts, initialization files, and datafiles to make the Oracle9i Enterprise Edition an Oracle9i Real Application Clusters database.

### Oracle system identifier (*sid*)

The Oracle System Identifier (*sid*) identifies a specific instance of the running Oracle software. For an Real Application Clusters database, each **node** within the cluster has an instance referencing the database.

The database name, specified by the `DB_NAME` parameter in the `INITDB_NAME.ORA` file, and unique thread ID make up each node's *`sid`*. The thread ID starts at 1 for the first instance in the cluster, and is incremented by 1 for the next instance, and so on. See also: **instance name**.

### OSD

See **operating system-dependent (OSD) clusterware**.

### Oracle Universal Installer (OUI)

A tool to install the Oracle relational database software. You can also use the Oracle Universal Installer to launch the **Database Configuration Assistant (DBCA)**.

### parameter file (PFILE)

A file used by the Oracle server that provide specific values and configuration settings that are used on database startup. The keyword `PFILE` is used in the startup command.

### Performance Manager

See: **Oracle Performance Manager**.

### preferred credentials

Each **Oracle Enterprise Manager** administrator can set up specific user names, passwords, and roles for nodes, listeners, databases, and other services that you administer in the network.

### Real Application Clusters

See **Oracle Real Application Clusters**.

### Recovery Manager (RMAN)

An Oracle tool that enables you to back up, copy, restore, and recover datafiles, control files, and archived redo logs. It is included with the Oracle server and does not require separate installation. You can invoke RMAN as a command line utility from the operating system (O/S) prompt or use the GUI-based Enterprise Manager Backup Manager.

### repository database

A set of tables in an Oracle database that stores data required by **Oracle Enterprise Manager**. This database is separate from the database on the nodes.

### RMAN

See: **Recovery Manager (RMAN)**.

### scalability

The ability to add additional nodes to Real Application Clusters applications and achieve markedly improved scale-up and speed-up.

### seed database

A preconfigured, ready-to-use database that requires minimal user input to create. *S*ee also: **starter database**.

### server clustering

See **Real Application Clusters**.

### Server Control (SRVCTL) Utility

Server Management uses the SRVCTL utility (installed on each node) to manage configuration information that is used by some Oracle tools. For example, SRVCTL serves as a single point of control between the Oracle Intelligent Agent and the nodes. Only one node's Oracle Intelligent Agent is used to communicate with SRVCTL. SRVCTL on that **node** then communicates to the other nodes through Oracle Net.

### Server Management (SRVM)

Server Management (SRVM) comprises the components required to operate **Oracle Enterprise Manager** in Real Application Clusters. The SRVM components, such as the Intelligent Agent, Global Services Daemon, and SRVCTL, enable you to manage cluster databases running in heterogeneous environments through an open client/server architecture using Oracle Enterprise Manager.

### server parameter file

A binary parameter file that resides on the Oracle Server. This file contains parameter settings that are both global and instance-specific. These parameter settings are persistent across instance shutdown and instance startup events.

### service discovery

When you execute the Discover Node command from the **Console**, the **Management Server** contacts the **Oracle Intelligent Agent** installed on that node to discover the Oracle services installed on the node. The Management Server then places the new information in the repository and updates the hierarchical tree in the Navigator window of the **Console**, displaying a broad view of all nodes and their respective services.

### service name

A logical representation of a database. This is the way a database is presented to clients. A database can be presented as multiple services and a service can be implemented as multiple database instances. The service name is a string that is the global database name, a name comprised of the database name (`DB_NAME`) and domain name (`DB_DOMAIN`), entered during installation or database creation.

### service registration

A feature whereby PMON automatically registers information with a listener. Because this information is registered with the listener, the `listener.ora` file does not need to be configured with this static information.

### shared current (scur)

The buffer state name for shared access mode to a block.

### shared mode (s)

A protected read block access mode. No writes are allowed in shared mode. In shared mode, any number of users can have simultaneous read access to a resource. See also: **exclusive (X) access mode**.

### shared server

A server that is configured to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With shared server configurations, many user processes connect to a dispatcher.

### sid

See: **Oracle system identifier (sid)**.

### spfile.ora

The binary parameter file that resides on the Oracle Server.

### snapshot control file

RMAN creates a snapshot control file to resynchronize from a read-consistent version of the control file. This is a temporary snapshot 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.

### sqlnet.ora File

A configuration file for the client or server that specifies:

- Client domain to append to unqualified service names or net service names
- Order of naming methods the client should use when resolving a name
- Logging and tracing features to use
- Route of connections
- Preferred Oracle Names servers

- External naming parameters
- Oracle Advanced Security parameters

The sqlnet.ora file typically resides in the $ORACLE_HOME/network/admin directory on UNIX platforms and in the %ORACLE_HOME%\network\admin directory on Windows platforms.

### SRVCTL

See **Server Control (SRVCTL) Utility**.

### starter database

See: **seed database**.

### SYSDBA

A database administration role that contains all system privileges with the ADMIN OPTION, and the SYSOPER system privilege. SYSDBA also permits CREATE DATABASE and time-based recovery.

### SYSOPER

A database administration role that permits a database administrator to perform any of the following commands: STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG, and RECOVER. SYSOPER includes the RESTRICTED SESSION privilege.

### TAF

See: **transparent application failover (TAF)**.

### thread

Each Oracle instance has its own set of online redo log groups. These groups are called a thread of online redo. In non-Real Application Clusters environments, each database has only one thread that belongs to the instance accessing it. In Real Application Clusters environments, each instance has a separate thread, that is, each instance has its own online redo log. Each thread has its own current log member.

### thread number

The number of the redo thread to be used by an instance as specified by the THREAD initialization parameter or the THREAD clause in the ALTER DATABASE ADD LOGFILE statement. You can use any available redo thread number but an instance cannot use the same thread number as another instance.

### tnsnames.ora file

A file that contains net service names. This file is needed on clients, nodes, the **Console**, and the Oracle Performance Manager machine.

### transparent application failover (TAF)

A runtime failover for high-availability environments, such as Real Application Clusters and Oracle Real Application Clusters Guard, TAF refers to the failover and re-establishment of application-to-service connections. It enables client applications to automatically reconnect to the database if the connection fails, and optionally resume a SELECT statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI) library.

### Virtual Interface Architecture (VIA)

A memory-based networking interface.

### X

See: **exclusive (X) access mode**.

# Index