

Oracle9i Real Application Clusters

Setup and Configuration

Release 2 (9.2)

March 2002

Part No. A96600-01

Oracle9i Real Application Clusters Setup and Configuration, Release 2 (9.2)

Part No. A96600-01

Copyright © 1998, 2002 Oracle Corporation. All rights reserved.

Primary Author: Mark Bauer.

Primary Contributors: David Austin, Jack Cai, Jonathan Creighton, and Raj Kumar.

Contributors: Sashikanth Chandrasekaran, Harvey Eneman, Mitch Flatland, Carmen Frank, Peter Povinec, Deborah Steiner, and Klaus Thielen.

Graphic Designer: Valarie Moore.

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

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

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark, and Oracle9i, Oracle8i, Oracle Names, SQL*Plus, Oracle Store, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	xiii
Preface.....	xv
What's New in Real Application Clusters Setup and Configuration?.....	xxvii
Part I Setup and Configuration of Oracle9i Real Application Clusters	
1 Introduction to Real Application Clusters Installation and Configuration	
Real Application Clusters Software Components	1-2
The Oracle9i Enterprise Edition and Oracle9i Real Application Clusters	1-2
The Operating System-Dependent Clusterware.....	1-2
The Oracle Universal Installer and Real Application Clusters.....	1-3
The Installed Real Application Clusters Storage Components.....	1-5
Installation Overview	1-5
System Installation Requirements	1-5
Hardware and Software Requirements for Oracle9i Real Application Clusters.....	1-6
Supported Interconnect Software.....	1-7
Version Compatibility.....	1-7
2 Configuring the Shared Disks	
Configuring Shared Disk Subsystems for Real Application Clusters	2-2
The Configuration Raw Device	2-2
Additional Raw Devices Required by the DBCA	2-3

	Recommended Tablespace and File Capacities.....	2-5
	Configuring Logical Devices.....	2-7
	Configuring Raw Volumes for Real Application Clusters on UNIX.....	2-7
	Configuring Logical Drives on Windows NT and Windows 2000	2-12
3	Using the Oracle Universal Installer for Real Application Clusters	
	Starting the Oracle Universal Installer.....	3-2
	Selecting a Database Configuration Type	3-5
	Database Configuration Assistant Processing.....	3-8
	Using the Pre-Configured Database Configuration Types.....	3-8
	Using the Customized Database Configuration Type	3-9
	Deinstalling Real Application Clusters with the Oracle Universal Installer	3-10
4	Creating Databases with the Database Configuration Assistant	
	Using the Database Configuration Assistant.....	4-2
	Benefits of Using the Database Configuration Assistant	4-2
	Creating the Database after Installation Using the Database Configuration Assistant.....	4-3
	Creating a Real Application Clusters Database with the DBCA.....	4-3
	Deleting a Database with the Database Configuration Assistant	4-19
	Migrating or Upgrading to Release 2 (9.2)	4-23
	Co-Existence of Oracle Versions on the Same Cluster	4-23
	Co-Existence of Operating System-Dependent Clusterware on Windows Platforms	4-23
	Rolling Upgrades	4-23
	Multiple Oracle Homes	4-23
5	Manually Creating Real Application Clusters Databases	
	Setting CREATE DATABASE Options for Cluster-Enabled Environments	5-2
	Setting MAXINSTANCES	5-2
	Setting MAXLOGFILES and MAXLOGMEMBERS.....	5-2
	Setting MAXLOGHISTORY	5-2
	Setting MAXDATAFILES	5-3
	Setting ARCHIVELOG Mode	5-3
	Changing Values for CREATE DATABASE Options.....	5-4
	Database Objects to Support Clusters	5-4
	Creating Additional Undo Tablespaces	5-4
	Configuring the Online Redo Log for Real Application Clusters	5-7

Creating a Database Manually	5-9
Install Oracle Products.....	5-10
Manually Create the Database.....	5-10
Creating Real Application Clusters Data Dictionary Views with CATCLUST.SQL.....	5-20

6 Configuring the Server Parameter File in Real Application Clusters Environments

Parameter Files and Real Application Clusters	6-2
Using The Server Parameter File in Real Application Clusters	6-2
Location of The Server Parameter File	6-2
Parameter File Search Order.....	6-5
Migrating to the Server Parameter File in Real Application Clusters Environments.....	6-5
Server Parameter File Placement in Real Application Clusters.....	6-5
Procedures for Migrating to the Server Parameter File	6-5
Server Parameter File Errors	6-6
Backing Up the Server Parameter File	6-7

Part II Oracle Enterprise Manager Installation Highlights for Real Application Clusters

7 Configuration Highlights for Using Enterprise Manager in Real Application Clusters Environments

Server Management Architecture	7-2
Server Management Requirements.....	7-4
Understanding the Oracle Enterprise Manager Setup	7-5
Using the Console, Management Server, and Repository on the Same Machine.....	7-5
Using the Console, Management Server, and Repository on Separate Machines	7-6
Notes for Running the Console in Standalone Mode.....	7-7
Configuring Enterprise Manager for Real Application Clusters Management.....	7-8
Task 1: Oracle Enterprise Manager Installation.....	7-8
Task 2: Specify Preferred Credentials for Nodes and Database	7-8
Task 3: Create an Operating System Account	7-9
Task 4: Grant SYSDBA or SYSOPER Privileges to a Database User	7-9
Task 5: Set Preferred Credentials in the Console	7-10
Using Oracle Performance Manager for Real Application Clusters Management.....	7-11

Task 1: Start Oracle Performance Manager.....	7-11
Task 2: Accessing Statistical Charts.....	7-11

Part III The Real Application Clusters Installed Configuration and High Availability

8 Understanding the Installed Configuration for Real Application Clusters

Understanding the Configured Environment.....	8-2
Shared Configuration File	8-2
Executing srvconfig on UNIX Platforms.....	8-2
Executing srvconfig on Windows Platforms	8-3
UNIX Operating System Configurations.....	8-3
Database Components Created Using the Database Configuration Assistant	8-4
Tablespaces and Datafiles.....	8-4
Redo Log Files	8-6
Control Files.....	8-7
Managing Undo Tablespaces.....	8-7
Initialization Parameter Files	8-7
Configuring Service Registration-Related Parameters in Real Application Clusters	8-8
Configuring the Listener File (listener.ora).....	8-9
Nondefault Listeners.....	8-9
Multiple Listeners.....	8-9
How Oracle Uses the Listener (listener.ora File).....	8-10
Directory Server Access (ldap.ora File).....	8-12
Net Service Names (tnsnames.ora File)	8-12
Profile (sqlnet.ora File).....	8-16

9 Configuring High Availability Features for Real Application Clusters

Transparent Application Failover	9-2
FAILOVER_MODE Parameters.....	9-2
TAF Implementation	9-2
Primary/Secondary Instances	9-6
Overview of Primary and Secondary Instances.....	9-6
Initialization Parameter Configuration	9-7

Client Configuration	9-7
Listener Configuration	9-8
Connecting to Secondary Instances	9-8
Warming the Library Cache on the Secondary Instance	9-9
Real Application Clusters Guard II	9-9
Configuring Clients for Real Application Clusters	9-10
Implementation of Client Configurations.....	9-11
Testing the Oracle Net Configuration	9-12

Part IV Migrating to Real Application Clusters

10 Migrating to Real Application Clusters from Single-Instance Oracle Databases

Migrating from a Single Instance to Real Application Clusters	10-2
Deciding to Migrate.....	10-2
Preparing to Migrate	10-2
Migrating Databases from Single Instance to Cluster-Enabled Environments.....	10-3
Migrating with Raw Devices or Shared File Systems	10-7
Upgrading Oracle8i Configurations to Oracle9i Configurations	10-8

Part V Reference for Real Application Clusters Setup and Configuration

A Directory Structure for Real Application Clusters Environments

Understanding the Real Application Clusters Directory Structure.....	A-2
UNIX Directory Structures for Real Application Clusters	A-2
Windows Directory Structures for Real Application Clusters	A-3
Shared Oracle Home	A-4

B Troubleshooting Oracle Enterprise Manager in Real Application Clusters

Resolving Service Discovery Failures	B-2
Understanding Discovery	B-2
The UNIX oratab File and the Windows NT and Windows 2000 Registry	B-3
SRVM Configuration Repository	B-4
The listener.ora File	B-4

The tnsnames.ora File.....	B-5
The sqlnet.ora File.....	B-6
Discovery Results	B-6
Troubleshooting Discovery	B-7

Glossary

Index

List of Figures

1-1	Installation Components for Real Application Clusters.....	1-3
3-1	Oracle Universal Installer Cluster Node Selection Page	3-4
4-1	DBCA Welcome Page for Real Application Clusters.....	4-4
4-2	Database Configuration Assistant Operations Page.....	4-6
4-3	Database Configuration Assistant Node Selection Page.....	4-7
4-4	Database Configuration Assistant Database Templates Page.....	4-8
4-5	Database Configuration Assistant Database Identification Page.....	4-9
4-6	Database Configuration Assistant Database Features Page.....	4-10
4-7	Database Configuration Assistant Database Connections Options Page.....	4-11
4-8	Database Configuration Assistant Initialization Parameters Page.....	4-12
4-9	All Initialization Parameters Dialog.....	4-13
4-10	Database Configuration Assistant Database Storage Page.....	4-15
4-11	Database Configuration Assistant Creation Options Page.....	4-17
4-12	Database Configuration Assistant Summary Dialog.....	4-18
4-13	Database Configuration Assistant Operations Page.....	4-20
4-14	Database Configuration Assistant List of Cluster Databases Page.....	4-21
4-15	Database Configuration Assistant Database Deletion Summary Dialog.....	4-22
6-1	File Locations Tab For Initialization Parameters.....	6-4
7-1	SRVCTL Architecture for UNIX and Windows NT and Windows 2000.....	7-3
7-2	Console, Management Server, and Repository on the Same Machine.....	7-6
7-3	Console, Management Server, and Repository on Separate Machines.....	7-7

List of Tables

2-1	Minimum Tablespace Sizes.....	2-5
2-2	Recommended Names for Oracle Database Files for Two-Instance Configurations.....	2-7
2-4	Symbolic Link Names for Preconfigured Database Types.....	2-13
3-1	Privilege Verification Methods.....	3-2
3-2	Oracle Universal Installer Database Configuration Types.....	3-6
5-1	Component Settings for Manual Database Creation.....	5-11
5-2	Database Name, Domain, Global Name, and SID Prefix Example Settings.....	5-11
5-3	Component Settings for Node and Host Name and Thread ID	5-12
5-4	Host and Node Name, Thread ID and SID	5-12
5-5	Oracle Net Configuration File Requirements.....	5-19
7-1	SRVM Instance Management Utilities	7-2
7-2	Node Requirements	7-4
8-1	Tablespace Names used by Oracle Real Application Clusters Databases	8-4
8-2	Windows NT and Windows 2000 Symbolic Link Names	8-5
8-3	Connections for Net Service Names	8-13
9-1	Listener.ora Example without sid_list_listener Entry	9-8
A-1	Directory Structure for A Sample OFA-Compliant UNIX Environment.....	A-2
A-2	Directory Structure for A Sample OFA-Compliant Windows Environment	A-3

Send Us Your Comments

Oracle9i Real Application Clusters Setup and Configuration, Release 2 (9.2)

Part No. A96600-01

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 Setup and Configuration explains how to install and configure **Real Application Clusters**. 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 navigate within the online Oracle9i Real Application Clusters Documentation set
- Your platform-specific installation guide for installation information about Oracle Real Application Clusters Guard

This preface contains these topics:

- [Audience](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)
- [Documentation Accessibility](#)

Audience

Oracle9i Real Application Clusters Setup and Configuration is primarily for network or Database Administrators (DBAs) responsible for the installation and configuration of Real Application Clusters.

To use this document you should have a conceptual understanding of Real Application Clusters processing and its software and hardware components as described in *Oracle9i Real Application Clusters Concepts*.

Organization

This document contains the following five parts:

Part I: Setup and Configuration of Oracle9i Real Application Clusters

Part I describes the Real Application Clusters installation procedures.

Chapter 1, "Introduction to Real Application Clusters Installation and Configuration"

This chapter describes the Real Application Clusters software installation process and the hardware and software requirements for Real Application Clusters.

Chapter 2, "Configuring the Shared Disks"

This chapter explains how to configure shared disk subsystems for Real Application Clusters in environments that do not support or for which you choose not to use a cluster file system.

Chapter 3, "Using the Oracle Universal Installer for Real Application Clusters"

This chapter explains how to use the Oracle Universal Installer to install the Oracle Enterprise Edition with the Real Application Clusters software.

Chapter 4, "Creating Databases with the Database Configuration Assistant"

This chapter explains how to use the Database Configuration Assistant to create Real Application Clusters databases.

Chapter 5, "Manually Creating Real Application Clusters Databases"

This chapter explains how to manually create Real Application Clusters databases.

Chapter 6, "Configuring the Server Parameter File in Real Application Clusters Environments"

This chapter explains how to configure and use the server parameter file for Real Application Clusters databases.

Part II: Oracle Enterprise Manager Installation Highlights for Real Application Clusters

Part II describes important topics you should be aware of when installing Oracle Enterprise Manager in Real Application Clusters environments.

Chapter 7, "Configuration Highlights for Using Enterprise Manager in Real Application Clusters Environments"

This chapter describes the Real Application Clusters-specific topics for installing Oracle Enterprise Manager.

Part III: The Real Application Clusters Installed Configuration and High Availability

Part III describes the configuration for Real Application Clusters environments. It also describes how to configure Oracle high availability features for Real Application Clusters.

Chapter 8, "Understanding the Installed Configuration for Real Application Clusters"

This chapter describes the installed configuration in detail.

Chapter 9, "Configuring High Availability Features for Real Application Clusters"

This chapter explains how to configure High Availability features for Real Application Clusters.

Part IV: Migrating to Real Application Clusters

Part IV provides information about migrating to Real Application Clusters.

Chapter 10, "Migrating to Real Application Clusters from Single-Instance Oracle Databases"

This chapter describes how to migrate to Real Application Clusters.

Part V: Reference for Real Application Clusters Setup and Configuration

Part V provides reference information for Real Application Clusters.

Appendix A, "Directory Structure for Real Application Clusters Environments"

This appendix describes the directory structure for the installed Real Application Clusters software on both UNIX and Windows NT and Windows 2000 platforms.

Appendix B, "Troubleshooting Oracle Enterprise Manager in Real Application Clusters"

This appendix explains troubleshooting issues for Oracle Enterprise Manager's Server Management Component in Real Application Clusters environments.

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 Administration*
- *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

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

Oracle9i Real Application Clusters Management

- *Oracle9i Real Application Clusters Administration*
- *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 on 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 .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.

Convention	Meaning	Example
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepuTil class implements these methods.
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>Uold_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

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

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

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

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>

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.	<pre>SELECT last_name, employee_id FROM employees;</pre> <pre>SELECT * FROM USER_TABLES;</pre> <pre>DROP TABLE hr.employees;</pre>
lowercase	<p>Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	<pre>SELECT last_name, employee_id FROM employees;</pre> <pre>sqlplus hr/hr</pre> <pre>CREATE USER mjones IDENTIFIED BY ty3MUJ9;</pre>

Conventions for Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	<pre>c:\winnt\"\"system32 is the same as</pre> <pre>C:\WINNT\SYSTEM32</pre>

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 <i>command prompt</i> in this manual.	C:\oracle\oradata>
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp scott/tiger TABLES=emp QUERY=\ "WHERE job='SALESMAN' and sal<1600\ C:\>imp SYSTEM/ <i>password</i> 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 OracleHOME_ NAMEtnsListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin95 for Windows 95 ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle.</p> <p>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 <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle9i Database Getting Started for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

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 Setup and Configuration?

This section describes the new features of Oracle9i release 2 (9.2) as they pertain to the setup and configuration of **Oracle Real Application Clusters**. The topic in this section is:

- [Oracle9i Release 2 New Features for Real Application Clusters Setup and Configuration](#)

See Also:

- Your platform-specific documentation for information about installing Oracle Real Application Clusters Guard I (except for Windows NT and Windows 2000 where Real Application Clusters Guard I installation information is in the Real Application Clusters Guard I document on the Fail Safe/Real Application Clusters Guard software CD-ROM)
- Your Real Application Clusters Guard II installation documentation on the Real Application Clusters Guard II software CD for information about installing Real Application Clusters Guard II

Oracle9i Release 2 New Features for Real Application Clusters Setup and Configuration

- **New Tablespace Requirements**

There are two new tablespaces required for this release: the XML tablespace to accommodate the XML DB features and the ODM tablespace to accommodate the Oracle Data Mining features.

- **Oracle Universal Installer Enhancements**

- The Installer propagates local inventory to the remote cluster nodes. Therefore, you can perform deinstalls and patch installs from any node in the cluster.
- Error messages are cluster-aware.
- Registry operations in the Installer are clusterized for Windows NT and Windows 2000 platforms.
- The order and composition of some Installer screens have changed and a Summary page displays the names of the nodes on which an operation is performed.
- On a machine where you are installing Oracle for the first time, the Installer asks you to run the root script for the creation of the `instLoc` file on all the cluster nodes.
- On UNIX platforms, the Installer displays a message asking you to run the `root.sh` script on the selected cluster nodes.

- **Database Configuration Assistant Enhancements**

- On some platforms you can use cluster file system files to store datafiles for your cluster database.
- You can specify the raw devices for datafiles in a text file and provide it as an argument to the Database Configuration Assistant. You can also place the SRVM Configuration Repository on a cluster file system.
- You can customize thread assignments to database objects. For example, you can customize the default undo tablespace-to-instance assignment and redo log group files-to-instance thread assignment on the DBCA Storage page.
- The DBCA displays an Instance Management summary page that shows the attributes of the instance to be added or deleted.

- You can install multiple instances on a single node.
- You can perform Instance Management operations, such as instance addition and deletion, using any active instance in your cluster database.
- You can use Instance Management in non-interactive silent mode so that all input is provided through command line arguments.
- When an operation fails, the DBCA reverses partially completed operations to restore the system to its previous state.
- **Server Management Architecture Enhancements**
 - You can use one `gsdctl` command syntax on all platforms to install, start, stop, and check the status of the GSD.
- **Database Upgrade Assistant Enhancements**
 - The Database Migration Assistant has been renamed the Database Upgrade Assistant (DUA). In addition, the DUA is fully functional to upgrade your pre-release 2 (9.2) Oracle cluster database to Oracle9i Real Application Clusters release 2 (9.2).
- **Dynamic Support for Local and Remote Listener Parameters**
 - You can use the `ALTER SYSTEM SET SQL` statement to dynamically update the `LOCAL_LISTENER` and `REMOTE_LISTENER` initialization parameters. Oracle also dynamically updates these parameters when you add or delete nodes from your cluster database. When Oracle updates these parameters, the PMON process registers the new information with the listener.

Part I

Setup and Configuration of Oracle9i Real Application Clusters

Part I begins with an overview of the components required to set up and configure Oracle9i Real Application Clusters and also provides procedures for configuration and installation of Oracle9i for Real Application Clusters. This part also describes database creation and parameter file configuration. The chapters in Part I are:

- Chapter 1, "Introduction to Real Application Clusters Installation and Configuration"
- Chapter 2, "Configuring the Shared Disks"
- Chapter 3, "Using the Oracle Universal Installer for Real Application Clusters"
- Chapter 4, "Creating Databases with the Database Configuration Assistant"
- Chapter 5, "Manually Creating Real Application Clusters Databases"
- Chapter 6, "Configuring the Server Parameter File in Real Application Clusters Environments"

Introduction to Real Application Clusters Installation and Configuration

This chapter provides an overview of the procedures and components for installing and configuring the **Real Application Clusters** software. This chapter includes the following topics:

- [Real Application Clusters Software Components](#)
- [Installation Overview](#)
- [System Installation Requirements](#)
- [Version Compatibility](#)

Real Application Clusters Software Components

Real Application Clusters software uses all the components of single instance Oracle environments plus cluster database software that facilitates internode communication. This section, with [Figure 1-1](#), briefly describes the process for installing this software.

The Oracle9i Enterprise Edition and Oracle9i Real Application Clusters

The Oracle9i Enterprise Edition, which is required for Real Application Clusters, provides single instance database software and the optional software to operate Real Application Clusters databases. Some of the Real Application Clusters-specific components include:

- Operating system-dependent clusterware
- Real Application Clusters-enabled `ORACLE_HOME`
- Oracle Server Management (SRVM) installs with the Real Application Clusters software

The Operating System-Dependent Clusterware

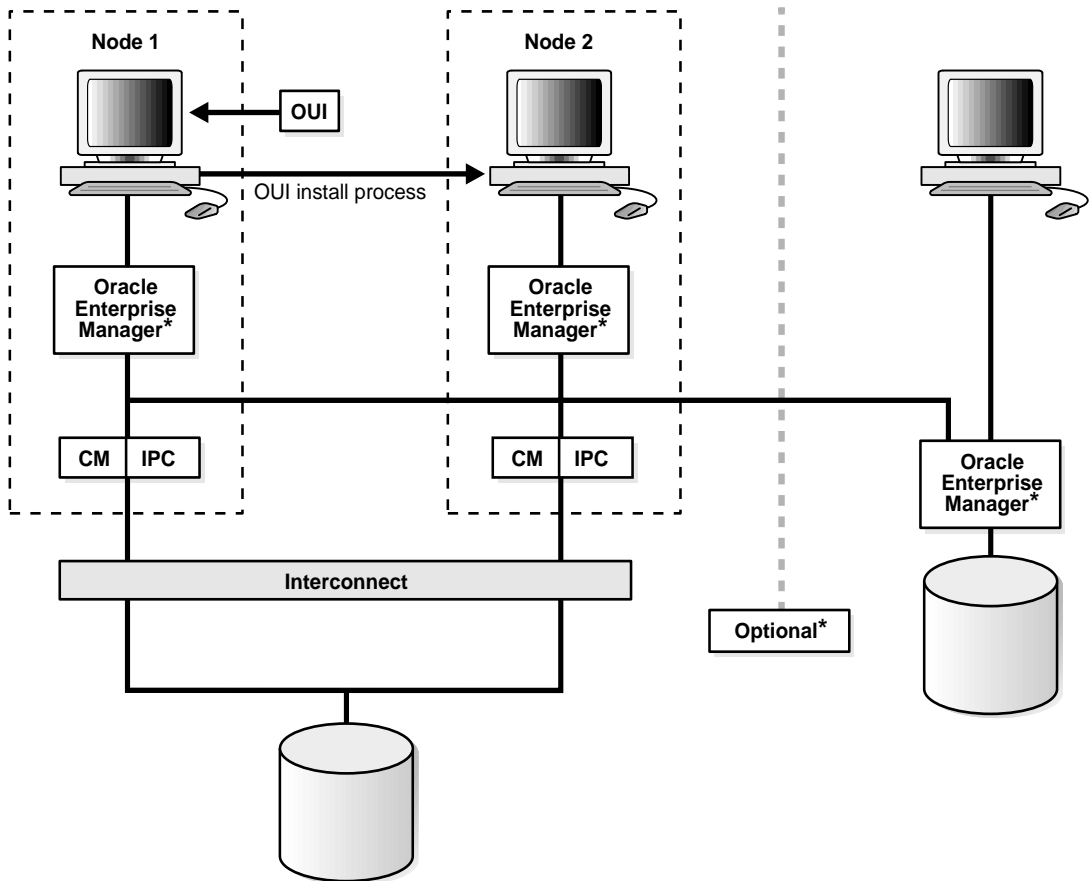
The operating system clusterware contains operating system-dependent (OSD) components that control the operating system and clusterware services required for Real Application Clusters. The OSD's two primary subcomponents are the Cluster Manager (CM) and the Interprocess Communication (IPC) software.

The OSD clusterware installation process varies according to platform. For UNIX operating systems, your vendor provides the OSD clusterware. For Windows NT and Windows 2000 operating systems, use the Oracle Cluster Setup Wizard to install the Oracle OSD clusterware after configuring your logical partitions as described in the online document, *Oracle9i Database Installation Guide for Windows*.

The Oracle Universal Installer and Real Application Clusters

The Oracle Universal Installer (OUI) provides a graphical user interface for the installation of the Oracle Enterprise Edition software. [Figure 1-1](#) shows the OUI process flow.

Figure 1-1 Installation Components for Real Application Clusters



When the OUI runs, Oracle Corporation recommends that you select one of the preconfigured databases or use the Database Configuration Assistant (DBCA) interactively to create your cluster database. You can also manually create your

database with scripts. If you are not using a cluster file system, then you must have configured shared raw devices before proceeding with database creation.

When you install Real Application Clusters, the installer copies the Oracle software onto the node from which you are running the installer. The OUI then propagates the software onto the other nodes that you have identified as part of your cluster database. You can also use the Installer to deinstall Oracle software.

The OUI can also install the software for the system management tool, Oracle Enterprise Manager (EM), to integrate this tool into your new cluster environment. Or you can install EM onto other client machines outside your cluster as shown in [Figure 1-1](#) (*).

The Oracle Intelligent Agent is installed when you install EM. The Oracle Intelligent Agent must be running on each node that is part of your Real Application Clusters database so that EM can discover and manage the database.

See Also:

- *Oracle9i Real Application Clusters Administration* for information about using Enterprise Manager to administer Real Application Clusters environments
- *Oracle9i Real Application Clusters Deployment and Performance* for information about using Performance Manager, which is part of Enterprise Manager, to monitor performance in Real Application Clusters environments
- *Oracle Intelligent Agent User's Guide* for information about the Oracle Intelligent Agent

The Installed Real Application Clusters Storage Components

All instances in Real Application Clusters environments share the control file, server parameter file, and all datafiles. These files reside on a shared cluster file system or on shared disks; either of these types of file configurations are accessed by all the cluster database instances. Each instance also has its own discrete set of redo log files. During failures, this arrangement of shared access to redo log files enables surviving instances to perform recovery.

Installation Overview

The Real Application Clusters installation process includes four major tasks.

1. Install the operating system-dependent (OSD) clusterware. The OSD clusterware installation process varies according to platform:
 - For UNIX, refer to your vendor documentation.
 - For Windows NT and Windows 2000, use the Oracle Cluster Setup Wizard as described in the online document, *Oracle9i Database Installation Guide for Windows*.
2. If you are not using a cluster file system, then configure the shared disks as described in [Chapter 2](#).
3. Run the OUI to install the Oracle9i Enterprise Edition and the Oracle9i Real Application Clusters software as described in [Chapter 3](#).
4. Create and configure your database as described in either:
 - [Chapter 4](#) that explains how to use the Database Configuration Assistant (DBCA), or:
 - [Chapter 5](#) that explains how to manually create your database

System Installation Requirements

Verify that your system meets the requirements discussed under the following topics before proceeding with installation:

- [Hardware and Software Requirements for Oracle9i Real Application Clusters](#)
- [File System Requirements for Using Oracle Managed Files \(Optional\)](#)

Hardware and Software Requirements for Oracle9i Real Application Clusters

Verify the following hardware and software requirements for each node:

Hardware

Each node in a cluster requires the following hardware:

- Operating system-specific hardware as described in operating system-specific installation guides
- External shared hard disk(s)

Software

Each node in a cluster requires the following software:

- Operating system-specific software as described in operating system-specific installation guides
- Certified vendor-supplied operating system-dependent clusterware for UNIX, or Oracle operating system-dependent clusterware for Windows NT and Windows 2000
- Oracle9i Enterprise Edition
- Oracle Net
- Oracle Intelligent Agent release 2 (9.2) if using EM
- One of the following Web browsers to view online documentation:
 - Netscape Navigator Version 3.0 or later.
<http://www.netscape.com>
 - Microsoft Internet Explorer Version 3.0 or later.
<http://www.microsoft.com>

File System Requirements for Using Oracle Managed Files (Optional) To use Oracle Managed Files (OMF), then your platform must support a cluster file system. The OMF feature automatically creates and deletes files that Oracle requires to manage the database.

See Also: *Oracle9i Database Administrator's Guide* for more information about Oracle Managed Files

Supported Interconnect Software

Interconnects that support Real Application Clusters and Cache Fusion use one of the following protocols:

- Transmission Control Protocol/Interconnect Protocol (TCP/IP)
- User Datagram Protocol (UDP)
- Virtual Interface Architecture (VIA)

Real Application Clusters can use any interconnect product that supports these protocols. You can also use other proprietary protocols that are hardware vendor-specific. The interconnect product that you use must be certified by Oracle for your Oracle9i Real Application Clusters hardware platform.

See Also: *Oracle9i Real Application Clusters Concepts* for more information about Cache Fusion

Version Compatibility

32-bit instances cannot share the same 9.2 database with 64-bit instances. Depending on the platform, for example, 32-bit and 64-bit Oracle9i Real Application Clusters executables cannot be used at the same time within the same cluster domain. This is because some cluster manager implementations may not be able to concurrently handle 32-bit and 64-bit clients.

If you have installed and configured the required hardware and software and you are not using a cluster file system, then proceed to [Chapter 2](#) to configure the shared disks. If you are using a cluster file system, then proceed to [Chapter 3](#).

Configuring the Shared Disks

This chapter describes how to configure shared disk subsystems to deploy Real Application Clusters in environments that do not support (or for which you do not want to use) cluster file systems (CFS). If your system supports a cluster file system and you do not want to use raw devices as described in this chapter, then proceed to [Chapter 3](#). The topics in this chapter include:

- [Configuring Shared Disk Subsystems for Real Application Clusters](#)
- [Configuring Raw Volumes for Real Application Clusters on UNIX](#)
- [Configuring Logical Drives on Windows NT and Windows 2000](#)

Configuring Shared Disk Subsystems for Real Application Clusters

Real Application Clusters requires that each instance share a set of unformatted devices on a shared disk subsystem if you are not using a cluster file system for datafiles.

The Oracle instances in Real Application Clusters write data onto the shared files to update the control file, server parameter file, each datafile, and each redo log file. The number and type of raw devices required depends on several factors as described in the following sections.

You must configure at least one device for configuration information as described under the following heading, "[The Configuration Raw Device](#)". If you select one of the preconfigured database options on the Installer's Database Configuration screen, or if you use the Database Configuration Assistant (DBCA) interactively, then you must also configure additional raw devices as described under the heading "[Additional Raw Devices Required by the DBCA](#)" on page 2-3.

The Configuration Raw Device

You must create at least one shared raw device as an information repository for the database server configuration. This device is referred to as the Server Management (SRVM) configuration device which is a component of Real Application Clusters. Enterprise Manager (EM) uses SRVM to perform Real Application Clusters-specific operations.

The SRVM core includes the server configuration file or raw device, the Global Services Daemon (GSD) which manipulates the device, and the Server Control (SRVCTL) Utility and interfaces to the configuration. SRVM also includes facilities that are used by the DBCA, the Oracle Universal Installer (OUI), and the Network Configuration Assistant (NetCA), to enable them to operate on multiple nodes.

You must create a shared raw device for SRVM:

- If this installation is the first Oracle9i Server installation on your cluster
- To migrate from a pre-release 1 (9.0.1) Oracle9i Real Application Clusters database to Oracle9i Release 2 (9.2)
- To manually create your database

Oracle Enterprise Manager (EM), the Database Configuration Assistant (DBCA), the Server Control (SRVCTL) Utility, and other management tools use this raw device to store configuration information about:

- The instances in your Real Application Clusters database

- The node that each instance runs on
- The Oracle home within which the database runs

Note: In Oracle8i and earlier releases, Oracle stored this information in either the `db_name.conf` file on UNIX platforms, or in the Registry on Windows NT platforms.

The configuration procedures that you use for this raw device depends on your operating system:

- On UNIX, configure a shared raw device for the Server Management (SRVM) configuration; the OUI prompts you for the name of this shared file.
- On Windows NT and Windows 2000, configure a Cluster Manager (CM) voting disk/SRVM Configuration Disk. Label this raw partition, `srvcfg`, by creating an Object Link Manager (OLM) symbolic link with a fixed name.

Note: On Windows NT and Windows 2000, SRVM uses the same raw partition as the Cluster Manager voting disk.

Configure additional raw devices as described in the next section to use the DBCA to create your database. The procedures for using the DBCA appear in [Chapter 4](#).

Additional Raw Devices Required by the DBCA

To use the **Database Configuration Assistant**, you must configure raw devices as described in this section. These devices are in addition to the configuration device mentioned in the previous section. Create these devices before running the OUI to install the Oracle Enterprise Edition software. The DBCA cannot create a Real Application Clusters database unless you have properly configured the following raw devices:

- Seven raw devices for each datafile
- At least two raw devices for control files
- One raw device for each instance for its own tablespace for automatic undo management, or one rollback segment tablespace to use manual undo management
- At least two for redo log files for each instance

- One for the server parameter file and five additional devices, one for each of the following features: Context, Oracle9i *interMedia*, OLAP (Online Analytical Processing), XML (Extensible Markup Language), and ODM (Oracle Data Mining)

See Also: The *Oracle9i Database Installation Guide for Windows* for additional information about raw device requirements for Windows NT and Windows 2000

Note: Each instance of a Real Application Clusters database has its own redo log files, but all instances in a cluster share the control files and datafiles. However, each instance's log files must be readable by all other instances for recovery.

Planning Your Raw Device Creation Strategy

Before installing the Oracle software, create enough partitions of specific sizes to support your database and leave a few spare partitions of the same size for future expansion. For example, if you have space on your shared disk array, select a limited set of standard partition sizes for your entire database. Partition sizes of 50MB, 100MB, 500MB, and 1GB are suitable for most databases. Also create a few very small and very large spare partitions that are, for example, 1MB and perhaps 5GB or greater in size. Based on your plans for using each partition, determine the placement of these spare partitions by combining different sizes on one disk, or by segmenting each disk into same-sized partitions.

Note: Ensuring that there are spare partitions enables you to perform emergency file relocations or additions if a tablespace datafile becomes full.

DBCA Database Configuration Options

The DBCA has four database templates, **General Purpose**, **Transaction Processing**, **Data Warehouse**, and **New Database**. [Chapter 3](#) describes these configuration templates in more detail. The first three templates contain preconfigured database options. The New Database template does not contain preconfigured options and is fully customizable.

If you use the General Purpose, Transaction Processing, or Data Warehouse configuration, then you must create specific tablespaces using the minimum sizes as

described under the heading "[Recommended Tablespace and File Capacities](#)" on page 2-5.

You can customize these tablespaces if you select the **Customized** database configuration type on the Universal Installer's Database Configuration screen. You can also specify the datafile file name and its size. To use the Customized configuration option, ensure that the raw volumes on UNIX, or raw partitions on Windows NT and Windows 2000, have enough space to accommodate the customized sizes.

Special Considerations for UNIX: Using Veritas Volume Manager

Use Veritas Volume Manager to create your volumes and then change the permissions and ownership on the volumes to the correct user, for example, `oracle` user. Also make sure that all volumes on the disks have the same names and the same permissions on each node. More information about Veritas Volume Manager is available at:

<http://www.veritas.com>

Recommended Tablespace and File Capacities

Use the tablespace size requirements in [Table 2-1](#) for the General Purpose, Transaction Processing, and Data Warehouse configuration types. These requirements apply to UNIX, Windows NT, and Windows 2000 platforms. If you use the New Database configuration type, then use these recommended sizes as guidelines.

Table 2-1 *Minimum Tablespace Sizes*

Create a Raw Device For	With File Size
SYSTEM tablespace	400MB
server parameter file	5MB
USERS tablespace	120MB
TEMP tablespace	100MB
UNDOTBS1 tablespace	250MB

The DBCA uses automatic undo management by default and you should have one undo tablespace for each instance.

Note: Undo tablespace raw device datafiles for all the preconfigured database templates should be at least 250MB in size.

Table 2-1 (Cont.) Minimum Tablespace Sizes

Create a Raw Device For	With File Size
EXAMPLE tablespace	160MB
CWMLITE tablespace	100MB
XML tablespace	50MB
ODM tablespace	250MB
INDX tablespace	70MB
TOOLS tablespace	12MB
DRSYS tablespace	250MB
First control file	110MB
Second control file	110MB
Two redo log files for each instance	120MB (for each file)
srvcfg for the SRVM Configuration Repository	100MB

Note: If you use manual undo management, then make the RBS tablespace raw device datafile at least 500MB in size.

Note: To expand a datafile after your Real Application Clusters database is in use, refer to your platform-specific documentation. To enable Oracle to automatically expand the datafile to the size permitted by your operating system, execute the ALTER DATABASE *datafile_name* AUTOEXTEND ON MAXSIZE *nM* statement, where *datafile_name* is the name of the file and *n* megabytes is the maximum size.

See Also: The README for the latest file size recommendations

Some operating systems require additional overhead for the partition sizes in [Table 2-1](#). Refer to your operating system-specific documentation for the exact raw partition size requirements.

If you do not use the DBCA and instead create your database manually, then the number of raw devices you create depends on the number of instances and database options that you install.

See Also: [Chapter 5, "Manually Creating Real Application Clusters Databases"](#) for more information about manual database creation

Configuring Logical Devices

The configuration of raw devices is operating system-specific as described in the following sections:

- [Configuring Raw Volumes for Real Application Clusters on UNIX](#) on page 2-7
- [Configuring Logical Drives on Windows NT and Windows 2000](#) on page 2-12

Refer to the section that corresponds to your operating system.

Configuring Raw Volumes for Real Application Clusters on UNIX

Use the following procedures to configure raw volumes on UNIX:

1. Obtain root privileges.

See Also: *Oracle9i Administrator's Reference* for your UNIX operating system

2. Oracle Corporation recommends that you create the datafiles, control files, and redo log files using the file name format in [Table 2-2](#). Although you can use any file naming scheme, the format in [Table 2-2](#) simplifies administration. This format uses file names that identify the database with a *db_name* entry and that identify the raw volume type by using the object names.

Table 2-2 Recommended Names for Oracle Database Files for Two-Instance Configurations

Database Object	Example Raw Volume
SYSTEM tablespace	<i>db_name_raw_system_400m</i>
Server parameter file	<i>db_name_raw_sfile_5m</i>
USERS tablespace	<i>db_name_raw_users_120m</i>
TEMP tablespace	<i>db_name_raw_temp_100m</i>

Table 2–2 (Cont.) Recommended Names for Oracle Database Files for Two-Instance Configurations

Database Object	Example Raw Volume
UNDOTBS1 tablespace	<i>db_name_raw_undotbs1_250m</i>
UNDOTBS2 tablespace	<i>db_name_raw_undotbs2_250m</i>
EXAMPLE tablespace	<i>db_name_raw_example_160m</i>
CWMLITE (OLAP) tablespace	<i>db_name_raw_cwmlite_100m</i>
XML tablespace	<i>db_name_raw_xml_50m</i>
ODM (Oracle Data Mining) tablespace	<i>db_name_raw_odm_20m</i>
INDX tablespace	<i>db_name_raw_indx_70m</i>
TOOLS tablespace	<i>db_name_raw_tools_12m</i>
DRSYS tablespace	<i>db_name_raw_drsys_250m</i>
controlfile1	<i>db_name_raw_controlfile1_110m</i>
controlfile2	<i>db_name_raw_controlfile2_110m</i>
Two redo log files for each instance	<i>db_name_raw_thread_lognumber_120m</i>

Where *thread* is the thread ID of the instance and *number* is the log number, 1 or 2, of the instance.

Note: If you do not use automatic undo management, then substitute the `undotbs1` and `undotbs2` entries with the following entry to accommodate the RBS tablespace raw volume:

```
db_name_raw_rbs_500m
```

The following step is recommended for all configuration types. If you select the New Database configuration type, then you can enter the raw device names on the DBCA screen.

3. On the node from which you plan to run the OUI, create an ASCII file identifying the database-object-to-raw-device mapping as shown in [Table 2–3](#). The DBCA requires that this mapping file exists during database creation. When creating the ASCII file content for the objects, name them using the format:

```
database_object=raw_device_file_path
```


Where *database_object* represents a particular database object and *raw_device_file_path* is the path of the datafile for that object.

Table 2–3 UNIX ASCII File Contents for the Oracle Universal Installer

Database Object	Used For
system	SYSTEM tablespace datafile
spfile	Server parameter file
users	USERS tablespace datafile
temp	TEMP tablespace datafile
undotbs1	Undo tablespace 1 datafile
undotbs2	Undo tablespace 2 datafile
example	EXAMPLE tablespace datafile
cwmlite	CWMLITE (OLAP) tablespace datafile
xml	XML tablespace datafile
odm	Oracle Data Mining (ODM) tablespace datafile
indx	INDX tablespace datafile
tools	TOOLS tablespace datafile
drsys	DRSYS tablespace datafile
control1	Control file 1 tablespace datafile
control2	Control file 2 tablespace datafile
redo1_1	First redo log file for the first instance
redo1_2	Second redo log file for the first instance
redo2_1	First redo log file for the second instance
redo2_2	Second redo log file for the second instance

Note: You must create at least two redo log file entries for each instance. If you do not use automatic undo management, then substitute the entries for `undotbs1` and `undotbs2` with an entry for `rbs` to use manual undo management.

When you create the ASCII file, separate each database objects from its paths with an equal (=) sign as shown in [Example 2-1](#):

Example 2-1 Example UNIX ASCII File for a Two-Instance General Purpose Real Application Clusters Database Configuration

```
system=/dev/vx/rdisk/oracle_dg/clustdb_raw_system_400m
spfile=/dev/vx/rdisk/oracle_dg/clustdb_raw_spfile_5m
users=/dev/vx/rdisk/oracle_dg/clustdb_raw_users_120m
temp=/dev/vx/rdisk/oracle_dg/clustdb_raw_temp_100m
undotbs1=/dev/vx/rdisk/oracle_dg/clustdb_raw_undotbs1_250m
undotbs2=/dev/vx/rdisk/oracle_dg/clustdb_raw_undotbs2_250m
example=/dev/vx/rdisk/oracle_dg/clustdb_raw_example_160m
cwmLite=/dev/vx/rdisk/oracle_dg/clustdb_raw_cwmLite_100m
xml=/dev/vx/rdisk/oracle_dg/clustdb_raw_xml_50m
odm=/dev/vx/rdisk/oracle_dg/clustdb_raw_odm_20m
indx=/dev/vx/rdisk/oracle_dg/clustdb_raw_indx_70m
tools=/dev/vx/rdisk/oracle_dg/clustdb_raw_tools_12m
drsys=/dev/vx/rdisk/oracle_dg/clustdb_raw_dr_250m
control1=/dev/vx/rdisk/oracle_dg/clustdb_raw_controlfile1_110m
control2=/dev/vx/rdisk/oracle_dg/clustdb_raw_controlfile2_110m
redo1_1=/dev/vx/rdisk/oracle_dg/clustdb_raw_log11_120m
redo1_2=/dev/vx/rdisk/oracle_dg/clustdb_raw_log12_120m
redo2_1=/dev/vx/rdisk/oracle_dg/clustdb_raw_log21_120m
redo2_2=/dev/vx/rdisk/oracle_dg/clustdb_raw_log22_120m
```

Note: If you are not using automatic undo management, then substitute the entries for `undotbs1` and `undotbs2` with:

```
rbs=/dev/vx/rdisk/oracle_dg/clustdb_raw_rbs_500m
```

To enable Oracle to determine the raw device volume names, specify that Oracle use the ASCII file described in this section. Do this by setting the following environment variable where *filename* is the complete pathname of the ASCII file that contains the entries shown in [Example 2-1](#):

```
setenv DBCA_RAW_CONFIG filename
```

UNIX Preinstallation Steps

After configuring the raw volumes, perform the following steps prior to installation as `root` user:

Note: You cannot complete these preinstallation procedures until you configure your raw volumes.

1. Make sure you have an `osdba` group defined in the `/etc/group` file on all nodes of your cluster. To designate an `osdba` group name and group number and `osoper` group during installation, these group names must be identical on all nodes of your UNIX cluster that will be part of the Real Application Clusters database. The default UNIX group name for the `osdba` and `osoper` groups is `dba`.
2. Create an `oracle` user account on each node so that the `oracle` user:
 - a. Is a member of the `osdba` group
 - b. Is used only to install and update Oracle software
 - c. Has write permissions on remote directories
3. Create a mount point directory on each node to serve as the top of your Oracle software directory structure so that:
 - a. The name of the mount point on each node is identical to that on the initial node
 - b. The `oracle` account has read, write, and execute privileges
4. On the node from which you will run the OUI, set up user equivalence by adding entries for all nodes in the cluster, including the local node, to the `.rhosts` file of the `oracle` account, or the `/etc/hosts.equiv` file.
5. Exit the `root` account when you are done.
6. As `oracle` user, check for user equivalence for the `oracle` user by performing a remote login (`rlogin`) to each node in the cluster. If you are prompted for a password, then you have not given the `oracle` user the same attributes on all nodes. You must correct this because the OUI cannot use the `rmp` command to copy Oracle products to the remote node's directories without user equivalence.

Note: UNIX clusters also require an environment setup similar to single-instance Oracle environments. For these instructions and other operating system-specific Real Application Clusters preinstallation instructions, see the *Oracle9i Installation Guide* for your UNIX operating system.

After configuring your raw volumes, proceed to [Chapter 4](#) to install the Oracle9i Enterprise Edition software and to configure your Oracle9i Real Application Clusters database.

Configuring Logical Drives on Windows NT and Windows 2000

If you do not use a cluster file system for datafiles, then the datafiles, control files, and redo log files must reside on unformatted raw devices on Windows NT and Windows 2000 platforms. On Windows, these are more commonly referred to as *logical drives* that reside within **extended partitions**. The extended partitions point to raw space on the disks. To configure the logical drives, create multiple logical partitions using Windows NT Disk Administrator or Windows 2000 Disk Management.

Before creating the logical partitions, first create **extended partitions** that point to the raw space on the disk. Then create multiple logical partitions within the extended partitions and assign symbolic link names to them using the **Object Link Manager (OLM)**.

See Also: *Oracle9i Database Installation Guide for Windows* for further information about configuring logical drives for Real Application Clusters on Windows NT and Windows 2000

The DBCA General Purpose, Transaction Processing, and Data Warehouse database configuration types require the symbolic link names shown in the left-hand column of [Table 2-4](#) for a two-instance Real Application Clusters database. Oracle uses these names to map the tablespaces as shown in the sample file in [Example 2-1](#) on page 2-10.

Table 2-4 Symbolic Link Names for Preconfigured Database Types

Symbolic Link Name...	Used for...
<i>db_name_system</i>	SYSTEM tablespace
<i>db_name_spfile</i>	Server parameter file
<i>db_name_users</i>	USERS tablespace
<i>db_name_temp</i>	TEMP tablespace
<i>db_name_undotbs1</i>	First UNDO tablespace
<i>db_name_undotbs2</i>	Second UNDO tablespace
<i>db_name_example</i>	EXAMPLE tablespace
<i>db_name_cwmlite</i>	CWMLITE tablespace
<i>db_name_xml</i>	XML tablespace
<i>db_name_odm</i>	ODM tablespace
<i>db_name_indx</i>	INDX tablespace
<i>db_name_tools</i>	TOOLS tablespace
<i>db_name_drsys</i>	DRSYS tablespace
<i>db_name_control1</i>	First control file
<i>db_name_control2</i>	Second control file
<i>srvcfg</i>	SRVM Configuration Repository
<i>db_name_redo_thread_number</i>	Redo Log Files
Where <i>thread</i> is the thread ID of the instance and <i>number</i> is the log number (1 or 2) for the instance.	Each instance must have at least two redo log files. If the database name is <i>db</i> , then the link names for the first instance should be:
	<i>db_redo1_1</i> <i>db_redo1_2</i>
	The link names for the second instance should be:
	<i>db_redo2_1</i> <i>db_redo2_2</i>

Note: If you do not use automatic undo management, then to accommodate a rollback segment tablespace, instead of using `db_name_undo` and `db_name_undo`, create the following symbolic link:

`db_name_rbs`

To install the Oracle operating system-dependent clusterware, run the Oracle Cluster Setup Wizard. The Cluster Setup Wizard installs the Object Link Manager tool with which you create persistent symbolic links to the logical drives.

If you do not install the Oracle OSD clusterware, then copy the Object Link Manager software from the preinstall directory on the Oracle CD. Refer to the online document, the *Oracle9i Database Installation Guide for Windows* for further procedures for completing the configuration using OLM.

After configuring your logical drives, proceed to [Chapter 3](#) to install the Oracle9i Enterprise Edition software and to configure your Real Application Clusters database.

Using the Oracle Universal Installer for Real Application Clusters

This chapter describes how to use the **Oracle Universal Installer (OUI)** to install the **Oracle9i Enterprise Edition** and the **Real Application Clusters** software. The topics in this chapter include:

- [Starting the Oracle Universal Installer](#)
- [Using the Pre-Configured Database Configuration Types](#)
- [Using the Customized Database Configuration Type](#)
- [Deinstalling Real Application Clusters with the Oracle Universal Installer](#)

Note: If you are not using a cluster file system, then perform the procedures in this chapter only after configuring your disk subsystem as described in [Chapter 2](#). You must complete the procedures in this chapter before continuing with the procedures for using the **Database Configuration Assistant (DBCA)** in [Chapter 4](#).

Starting the Oracle Universal Installer

Follow these procedures to use the (OUI) to install the Oracle Enterprise Edition and the Real Application Clusters software.

1. Before running the OUI, from the **node** where you intend to run the Installer, verify that you have administrative privileges on the other nodes. Do this using one of the operating system-specific privilege verification methods in [Table 3-1](#).

Table 3-1 Privilege Verification Methods

Operating System	Verification Method
On UNIX	Perform a remote copy (<code>rcp</code>) to all the nodes, including the local node.
On Windows NT and Windows 2000	<p>Enter the following command for each node that will be part of the cluster database where <code>node_name</code> is the node name returned from the <code>lsnodes.exe</code> utility that is in the <code>preinstall_rac</code> directory on the installation CD:</p> <pre>NET USE \\node_name\C\$</pre> <p>Oracle Corporation recommends using the same user name and password on each node in a cluster or use a domain user name. If you use a domain user name, log on under a domain with a username and password that has administrative privileges on each node.</p>

Then for UNIX platforms only, set the environment variables as described in the following two steps. For all other platforms proceed to step 4.

2. If the installation you are about to perform creates the first `ORACLE_HOME` on your cluster for Real Application Clusters, then set the `SRVM_SHARED_CONFIG` environment variable to the name of the 100MB volume **raw device** or shared file that you set up for configuration storage by entering, for example:

```
setenv SRVM_SHARED_CONFIG shared_filename
```

If you are using a cluster file system, then make sure the partition on which this file resides has least 100MB of free space.

3. Set the `DBCA_RAW_CONFIG` environment variable. Do this, for example, by executing the following where `filename` is the name of the ASCII file containing the configuration information:

```
setenv DBCA_RAW_CONFIG filename
```

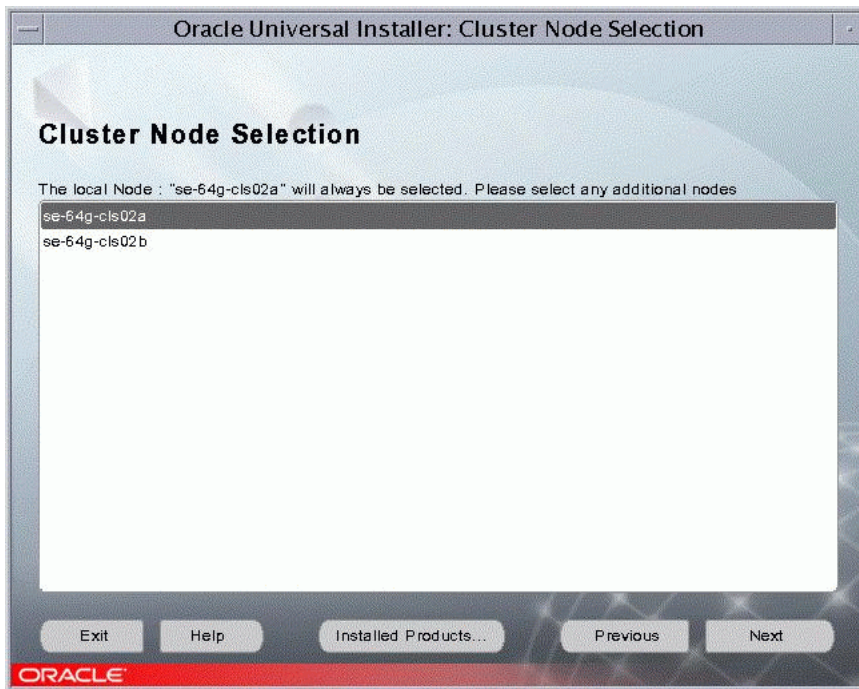

For the **Customized** configuration type, if this is the first installation on your cluster and you have not set the environment variable, then the Installer prompts you for the name of the 100MB file by displaying the Shared Configuration File Name page.

Note: All installations must be identical on all nodes in your cluster.

4. Run the Installer using one of the following procedures for your platform:
 - On UNIX, run the Installer 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. This page presents the OUI for Real Application Clusters. Click **Next**.

5. The Node Selection page appears as shown in [Figure 3-1](#).

Figure 3–1 Oracle Universal Installer Cluster Node Selection Page

6. If the Installer does not display the Node Selection page, then perform clusterware diagnostics by executing the `lsnodes -v` command and analyzing its output according to the following platform-specific instructions.
 - **UNIX Clusterware Diagnostics**

For UNIX platforms, execute the `lsnodes -v` command from the `/tmp/OraInstallYYYY-MM-DD_HH-MM-SSxx` directory, where `xx` is AM or PM).
 - **Windows NT and Windows 2000 Clusterware Diagnostics**

For these Windows platforms, execute the `lsnodes -v` command from the `preinstall_rac` directory.

For all platforms, refer to your vendor documentation if the output from the `lsnodes -v` command indicates that you have clusterware problems.
7. Select the nodes to include as part of your Real Application Clusters database. Note that the local node is always selected even though it may not be

highlighted on the Node Selection page. Click **Next**. The Installer displays the File Locations page.

8. On the File Locations page in the `Path` field under the `Destination` heading, enter the name of the Oracle home into which you are going to install the Oracle Enterprise Edition and Real Application Clusters software. Click **Next**. The Installer displays the Available Products page.
9. On the Available Products page, select **Oracle9i** and click **Next**. The Installer displays the Install Types page.
10. On the Install Types page, select **Enterprise Edition** and click **Next**. The Installer displays the Database Configuration page.

Before continuing, read the following section under the heading, "[Selecting a Database Configuration Type](#)" on page 3-5, which explains how the DBCA creates your database. This section also describes the configuration types and the underlying configuration that the DBCA creates.

However, if you are familiar with the DBCA and have already chosen a database configuration type, then:

11. On the Database Configuration page, select a configuration type and click **Next**. The Installer displays the Summary page which displays the names of all the products and the nodes onto which the Installer will load the Oracle Real Application Clusters software. While processing the installation, the Installer displays a progress message to indicate that the Installer is performing the remote installations.

Then proceed to one of the following sections:

- "[Using the Pre-Configured Database Configuration Types](#)" on page 3-8
- "[Using the Customized Database Configuration Type](#)" on page 3-9

Selecting a Database Configuration Type

When you run the OUI and select the Oracle9i Enterprise Edition, you can select the **General Purpose**, **Transaction Processing**, **Data Warehouse**, or **Customized** database configuration type for Real Application Clusters. For the first three configuration types, you must complete the procedures described later in this chapter. Then the Installer runs without further input.

If you select the Customized configuration in the Installer, then you can use the DBCA to create the database as described in [Chapter 4](#) or create the database manually as described in [Chapter 5](#). You can also select the Customized

configuration and use the DBCA to create a database using a customized version of one of the preconfigured templates. These templates correspond to the Installer's configuration types of General Purpose, Transaction Processing, and Data Warehouse. The DBCA can also use the **New Database** template to create a database.

To manually configure your environment, you can select the **Software Only** database configuration option. However, Oracle Corporation *strongly* recommends that you use one of the preconfigured database options or use the New Database option and the DBCA.

Configuration Type Descriptions

The configuration type that you select determines how you proceed. Table 3-2 describes the installation types.

Table 3-2 Oracle Universal Installer Database Configuration Types

Configuration Type	Description	Advantages
Data Warehouse, Transaction Processing, General Purpose	Installs a preconfigured starter database, licensable Oracle options (including Oracle9i Real Application Clusters), networking services, Oracle9i utilities, and online documentation. At the end of the installation, the DBCA creates a Real Application Clusters database and configures it for use.	Minimal input required. You can create your database more quickly than with the Customized type.
Customized	Enables you to customize your database options and storage components.	Enables you to create arbitrary tablespaces and datafiles and customize all aspects of your database.
Software Only	Installs only the software. Does not configure the listeners or network infrastructure and does not create a database.	—

The General Purpose, Transaction Processing, and Data Warehouse Configuration Types

If you select one of the first three configuration types on the Installer's Database Configuration screen, then complete the procedures in this chapter under the heading "[Using the Pre-Configured Database Configuration Types](#)" on page 3-8. These three configuration types use preconfigured templates. As mentioned, after you complete these procedures, the OracleNet Configuration Assistant and the DBCA run without further input and the Installer displays a progress indicator while it completes the installation.

The DBCA processing for these configuration types creates a starter database and configures the Oracle network services. As mentioned, if you do not use a cluster file system, then the DBCA verifies that you configured the raw devices for each tablespace.

Note: If you have not properly configured the raw devices as described in [Chapter 2](#), the DBCA cannot create your database.

If you select the Customized configuration, then you must enter specific information as described in the next section.

Using The Customized Configuration Type

If you select the Customized configuration type, then the OUI runs the DBCA which offers you a choice of four database templates:

- General Purpose
- Transaction Processing
- Data Warehouse
- New Database

The first three templates create a database that is optimized for that environment. You also can customize these templates. The New Database type, however, creates a database without using preconfigured options. Therefore, if you are not using a cluster file system, then you *must* enter specific raw device information for the datafiles to complete a New Database installation as described in [Chapter 4](#), "[Creating Databases with the Database Configuration Assistant](#)".

The following section provides more detail about DBCA processing when it creates a Real Application Clusters database.

Database Configuration Assistant Processing

The DBCA runs as part of the OUI installation process. When the Installer begins processing, the **Oracle Net Configuration Assistant** runs. If you configured directory access with the Oracle Net Configuration Assistant, then the DBCA will add an entry for the database service to the directory. Clients that you configured with directory access can also access the network information for the database service and connect to the database without a `tnsnames.ora` file.

After the Oracle Net Configuration Assistant completes its processing, the DBCA creates your database using the **optimal flexible architecture (OFA)**. This means that the DBCA creates your database files, including the default **server parameter file**, using standard file naming and file placement practices. The primary phases of DBCA processing are:

- Verify that you correctly configured the shared disks for each tablespace (for non-cluster file system platforms)
- Create the database
- Configure the Oracle network services
- Start the listeners and database instances

You can also use the DBCA in standalone mode to create a database.

See Also: The *Oracle9i Net Services Administrator's Guide* if you experience problems, for example, with the **listener** configuration, and for further information about **Lightweight Directory Access Protocol (LDAP)**-compliant support

Using the Pre-Configured Database Configuration Types

Continuing from Step 10 on page 3-5, execute the following procedures to complete a General Purpose, Transaction Processing, or Data Warehouse configuration:

1. On the Database Configuration page, select **General Purpose, Transaction Processing, or Data Warehouse**.
2. The Installer displays the Upgrading or Migrating an Existing Database page. Click **Next**.

See Also: *Oracle9i Database Migration* for information about using the Database Upgrade Assistant in Real Application Clusters environments

3. The Installer displays the Database Identification page on which you should enter a **global database name**. This is a name that comprises the database name and database domain, such as *db.us.acme.com*. Accept or change the common prefix for the Oracle **sid** for each **instance**.

Each instance has a *sid* that comprises the common prefix you enter in this step and a thread ID that is automatically generated. For example, if you enter *db* for the *sid*, the first database instance is given a *sid* of *db1*, and the second instance is given a *sid* of *db2*. Click **Next**.

4. The Installer displays the Database Character Set page on which you must select a character set and click **Next**.
5. The Installer displays a summary page. Review the components to be installed and click **Install** to begin the installation.

During installation, the installer displays progress indicators showing which components are installed.

When you complete these procedures, you are ready to administer and deploy your Real Application Clusters environment as described in *Oracle9i Real Application Clusters Administration* and *Oracle9i Real Application Clusters Deployment and Performance*.

Using the Customized Database Configuration Type

Continuing from Step 10 on page 3-5, execute the following procedures to complete a Customized configuration:

1. On the Database Configuration page, select **Customized**. The OUI displays the Node Selection page.
2. If the DBCA displays the Node Selection page, skip this step and proceed to step 3. Otherwise, perform the clusterware diagnostics as shown in the bulleted items in Step 6 on page 3-4.
3. Select the nodes that you want to include as part of your Real Application Clusters database. The local node is always selected even though it may not be highlighted on the Node Selection page. Click **Next**.
4. If you have not already set the raw device environment variable *and* if this is the first Oracle installation on this cluster, then the Installer displays the Shared Configuration File Name page. Enter the name of a raw device that is at least 100MB in size, or a cluster file system file on the Shared Configuration File Name page and click **Next**.

5. If the Installer detects a pre-existing Oracle database, then the Installer displays the Upgrading or Migrating an Existing Database page. Click **Next**.

See Also: *Oracle9i Database Migration* for information about using the Database Upgrade Assistant in Real Application Clusters environments

6. The Installer displays the Database Identification page on which you must enter a global database name. This is a name that comprises the database name and database domain, such as *db.us.acme.com*. Accept or change the common prefix that is to be used for the Oracle *sid* for each instance.

Each instance has a *sid* that comprises the common prefix that you enter in this step and a thread ID that is automatically generated. For example, if you enter *db* for the *sid*, the first database instance is given a *sid* of *db1*, and the second instance is given a *sid* of *db2*. Click **Next**.

The Installer displays the Database Character Set page on which you must select a character set and click **Next**.

7. The Installer displays a summary page. Review the components to be installed and click **Install** to begin the installation.

During installation, the installer displays progress indicators showing which components are installed.

When you complete these procedures, proceed to [Chapter 4, "Creating Databases with the Database Configuration Assistant"](#).

Deinstalling Real Application Clusters with the Oracle Universal Installer

Use the following procedures to deinstall the Oracle and Real Application Clusters software with the OUI:

1. Perform Steps 1 and 4 under the heading "[Starting the Oracle Universal Installer](#)" on page 3-2.
2. On the File Locations page in the `Path` field under the `Destination` heading, enter the name of the Oracle home from which you are going to deinstall the Oracle Enterprise Edition and Real Application Clusters software.
3. On the Available Products page, select **Deinstall** and click **Next**.

4. When Installer displays a deinstallation alert dialog, verify the information for the Oracle home from which the Installer will remove the Oracle software and click **OK**.

The Installer displays progress indicators as it completes the deinstallation process. The deinstall process deletes the Oracle software from all nodes in your cluster that have an `ORACLE_HOME` matching the entry from the File Locations page.

Creating Databases with the Database Configuration Assistant

This chapter describes using the **Database Configuration Assistant (DBCA)** to create and delete **Real Application Clusters** databases. It explains how to use the DBCA to create and delete Real Application Clusters databases. This chapter also briefly discusses upgrades, migrations, and multiple Oracle home issues. The topics in this chapter include:

- [Using the Database Configuration Assistant](#)
- [Benefits of Using the Database Configuration Assistant](#)
- [Creating the Database after Installation Using the Database Configuration Assistant](#)
- [Creating a Real Application Clusters Database with the DBCA](#)
- [Deleting a Database with the Database Configuration Assistant](#)
- [Migrating or Upgrading to Release 2 \(9.2\)](#)
- [Multiple Oracle Homes](#)

See Also: *Oracle9i Real Application Clusters Administration* for procedures on using the DBCA to add and delete instances

Using the Database Configuration Assistant

The DBCA creates your database using the **optimal flexible architecture (OFA)**. This means the DBCA creates your database files, including the default **server parameter file**, using standard file naming and file placement practices. The primary phases of DBCA processing are:

- If you did not use a cluster file system, then the DBCA verifies that you correctly configured the shared disks for each tablespace (for non-**cluster database** file system platforms)
- Create the database
- Configure the Oracle network services
- Start the listeners and database instances

See Also:

- ["Creating the Database after Installation Using the Database Configuration Assistant"](#) on page 4-3 for more information about using the DBCA in standalone mode
- The *Oracle9i Net Services Administrator's Guide* if you experience problems, for example, with the **listener** configuration, and for further information about **Lightweight Directory Access Protocol (LDAP)**-compliant directory support
- [Appendix A](#) for more information about Oracle Flexible Architecture

Benefits of Using the Database Configuration Assistant

Oracle Corporation recommends that you use the DBCA to create your database. This is because the DBCA preconfigured databases optimize your environment to take advantage of Oracle9i features such as the server parameter file and **automatic undo management**. The DBCA also enables you to define arbitrary tablespaces as part of the database creation process. Even if you have datafile requirements that differ from those offered in one of the DBCA templates, use the DBCA and modify the datafiles afterward. You can also execute user-specified scripts as part of the database creation process.

The DBCA and the **Oracle Net Configuration Assistant** also configure your Real Application Clusters environment for various Oracle high availability features and cluster administration tools.

To manually create your Real Application Clusters database, refer to [Chapter 5](#). The remainder of this chapter discusses using the DBCA to create a database.

Creating the Database after Installation Using the Database Configuration Assistant

If you did not create a database during installation, then you can create one later using the DBCA in standalone mode. To do this without a cluster file system, you must have configured each [raw device](#) as described in [Chapter 2](#). In addition, you must have configured your network using the OracleNet Configuration Assistant, or you can configure it manually. The [Global Services Daemon \(GSD\)](#) must also be running on each [node](#) in your cluster before you create the database in standalone mode.

If you use one of the Universal Installer's database configuration types or one of the DBCA's templates that uses preconfigured datafiles and if you do not have a cluster file system, then during database creation the DBCA first verifies that you created the raw devices for each tablespace. If you have not properly set up the raw devices, then you must replace the default datafiles with raw device names on the Storage page to continue database creation.

To start the DBCA, on one of the nodes:

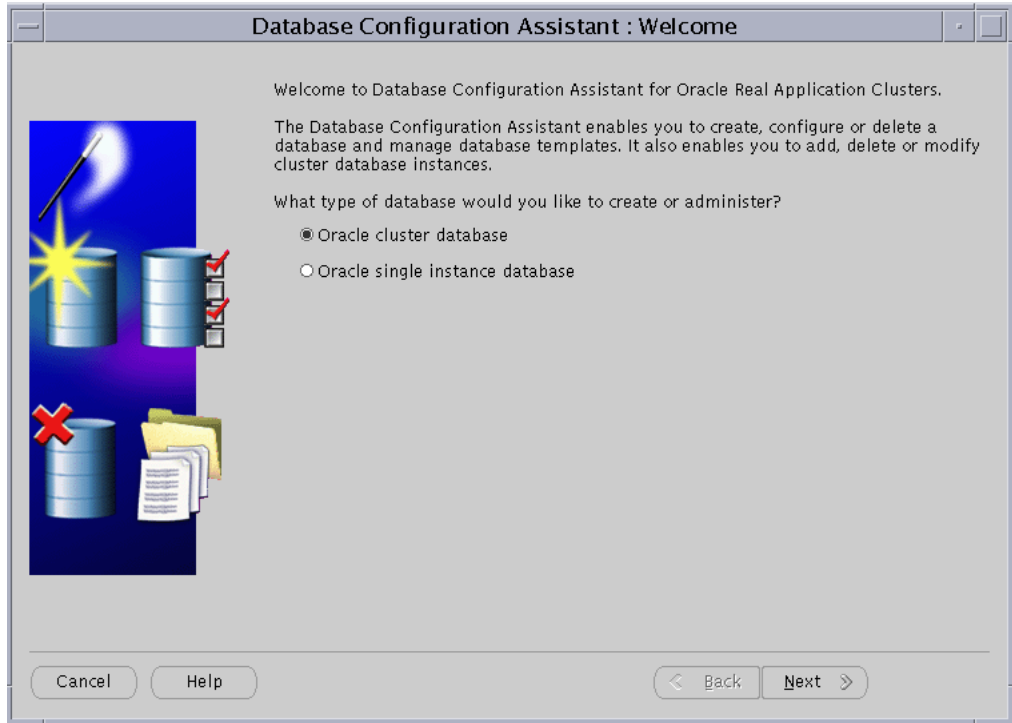
- On UNIX, enter the command `dbca` from the `$ORACLE_HOME/bin` directory
- On Windows NT and Windows 2000, choose Start > Programs > Oracle - `[HOME_NAME]` > Configuration and Migration Tools > Database Configuration Assistant

Creating a Real Application Clusters Database with the DBCA

The following section describes how to use the DBCA to create a database for Real Application Clusters. When you run the DBCA the first page displayed is the Database Configuration Assistant Welcome page for Real Application Clusters

shown in [Figure 4-1](#). The DBCA only displays this page when it detects that your **Cluster Manager (CM)** software is running.

Figure 4-1 DBCA Welcome Page for Real Application Clusters



See Also: The DBCA online help for more information

If the DBCA does not display the Real Application Clusters Welcome page with the Oracle cluster database option, then perform clusterware diagnostics as shown in the bulleted items in Step 6 on page 3-4.

Note: If the DBCA Welcome page for Real Application Clusters does not display, then it means that the DBCA was unable to:

- Detect whether the operating system-specific clusterware is running on UNIX platforms
 - Communicate with the Cluster Manager software on Windows NT and Windows 2000 platforms
-
-

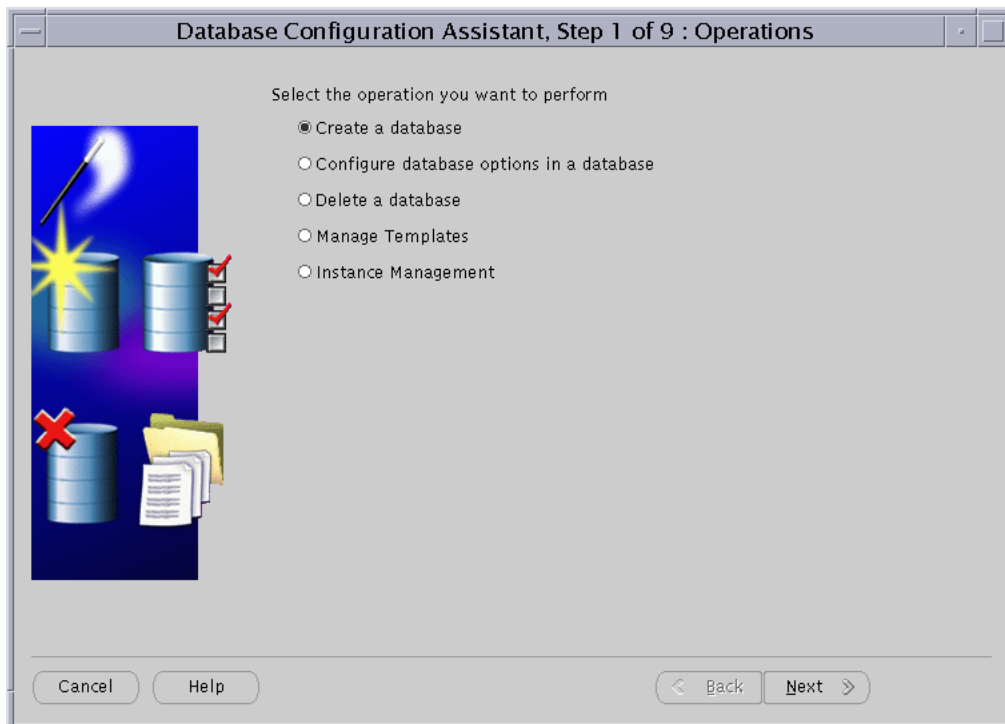
To create a Real Application Clusters database:

1. Select **Oracle cluster database** and click **Next**.

Warning: You cannot click **Back** after clicking **Next**. This prevents configuration problems that result from installing software for both single-instance and cluster database environments during the same DBCA session.

After you click **Next**, the DBCA displays the Operations page shown in [Figure 4-2](#).

Figure 4-2 Database Configuration Assistant Operations Page

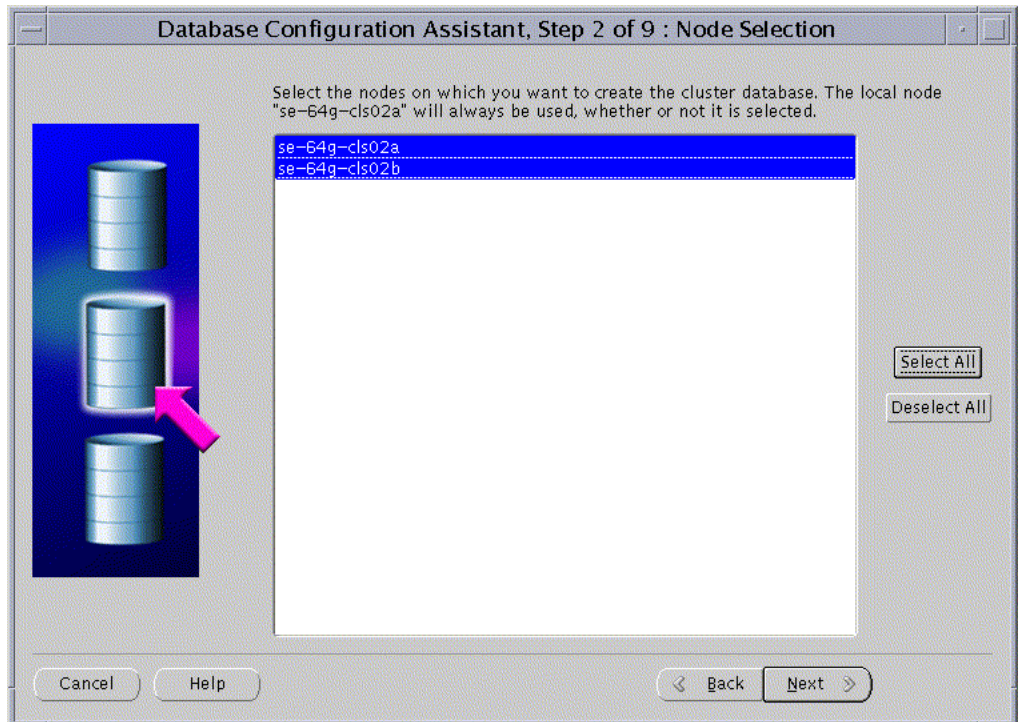


Note: The Instance Management option is only enabled if there is at least one Real Application Clusters database configured on your cluster.

2. Select **Create a database** and click **Next**.

After you click **Next**, The DBCA displays the Node Selection page shown in [Figure 4-3](#).

Figure 4-3 Database Configuration Assistant Node Selection Page



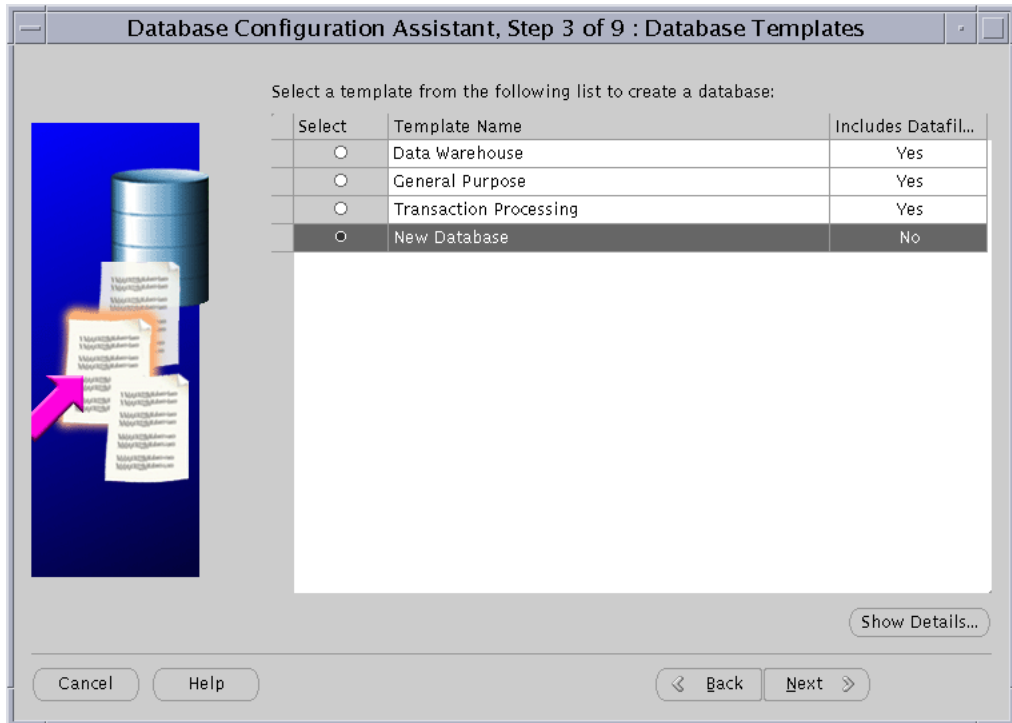
The Node Selection page shows the nodes that the DBCA detects in your cluster.

3. If nodes are missing from the Node Selection page, then perform clusterware diagnostics as under Step 6 on page 3-4.
4. Select the nodes that you want to configure as members of your cluster database and click **Next**. The DBCA always selects the local node by default.

If the GSD is not running on any of the nodes, then the DBCA displays a dialog explaining how to start the daemon using the `gsdctl start` command.

After you click **Next**, the DBCA displays the Database Templates page shown in [Figure 4-4](#).

Figure 4-4 Database Configuration Assistant Database Templates Page

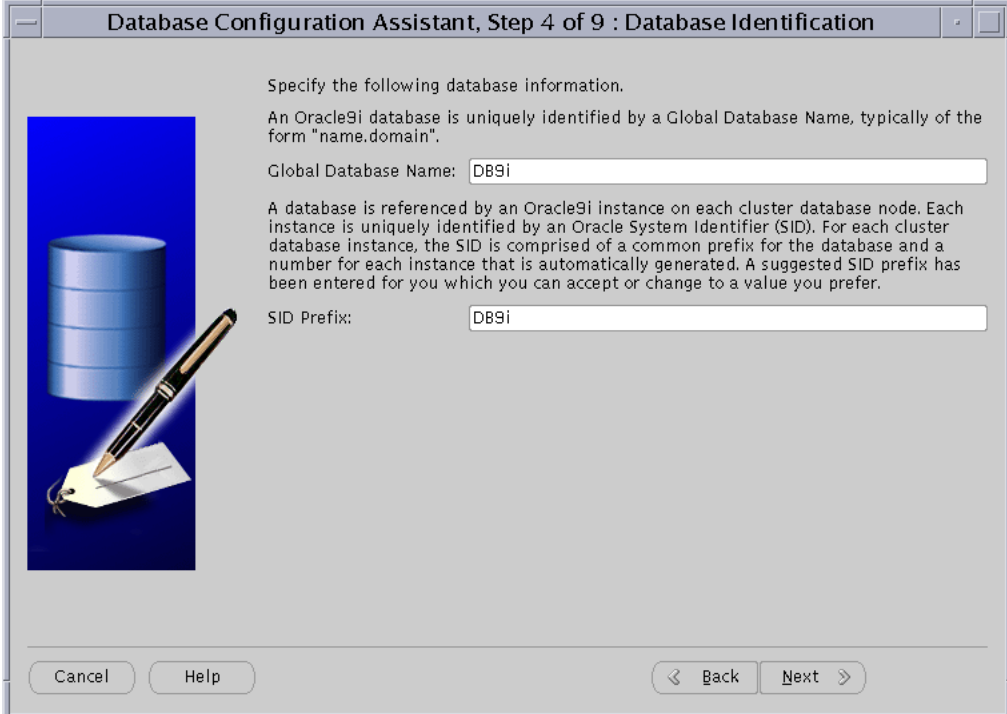


5. Select a template from which to create your cluster database and click **Next**. To create a preconfigured database, select a template that includes datafiles.

The templates on this page include the **Data Warehouse**, **Transaction Processing**, and **General Purpose** templates. These templates include datafiles and specially configured options for each environment. However, the **New Database** template does not include datafiles or the specially configured options.

After you click **Next**, the DBCA displays the Database Identification page shown in [Figure 4-5](#).

Figure 4-5 Database Configuration Assistant Database Identification Page



Database Configuration Assistant, Step 4 of 9 : Database Identification

Specify the following database information.

An Oracle9i database is uniquely identified by a Global Database Name, typically of the form "name.domain".

Global Database Name:

A database is referenced by an Oracle9i instance on each cluster database node. Each instance is uniquely identified by an Oracle System Identifier (SID). For each cluster database instance, the SID is comprised of a common prefix for the database and a number for each instance that is automatically generated. A suggested SID prefix has been entered for you which you can accept or change to a value you prefer.

SID Prefix:

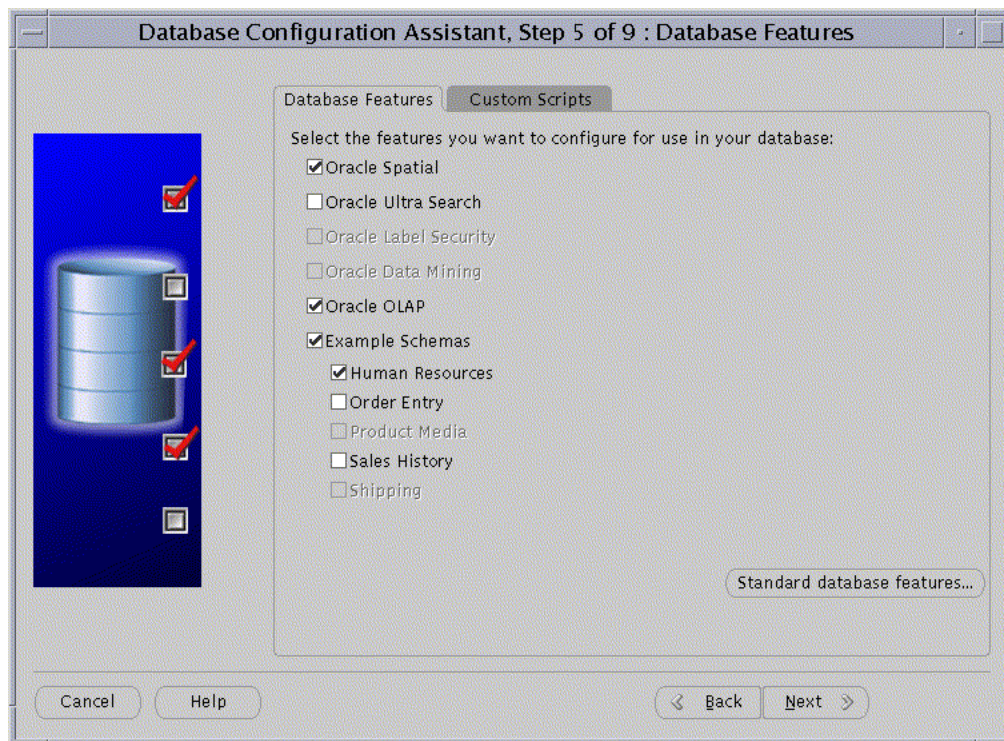
Cancel Help Back Next

6. Enter the **global database name** and the **Oracle system identifier (sid)** prefix for your cluster database and click **Next**.

Note: The SID prefix must be between one and five characters in length and the database name must begin with an alphabetical character.

After you click **Next**, if you have selected the **New Database** template, then the DBCA displays the Database Features page shown in [Figure 4-6](#). If you selected one of the other preconfigured database options, then after you click **Next** the DBCA displays the Initialization Parameters page shown in [Figure 4-8](#) on page 4-12.

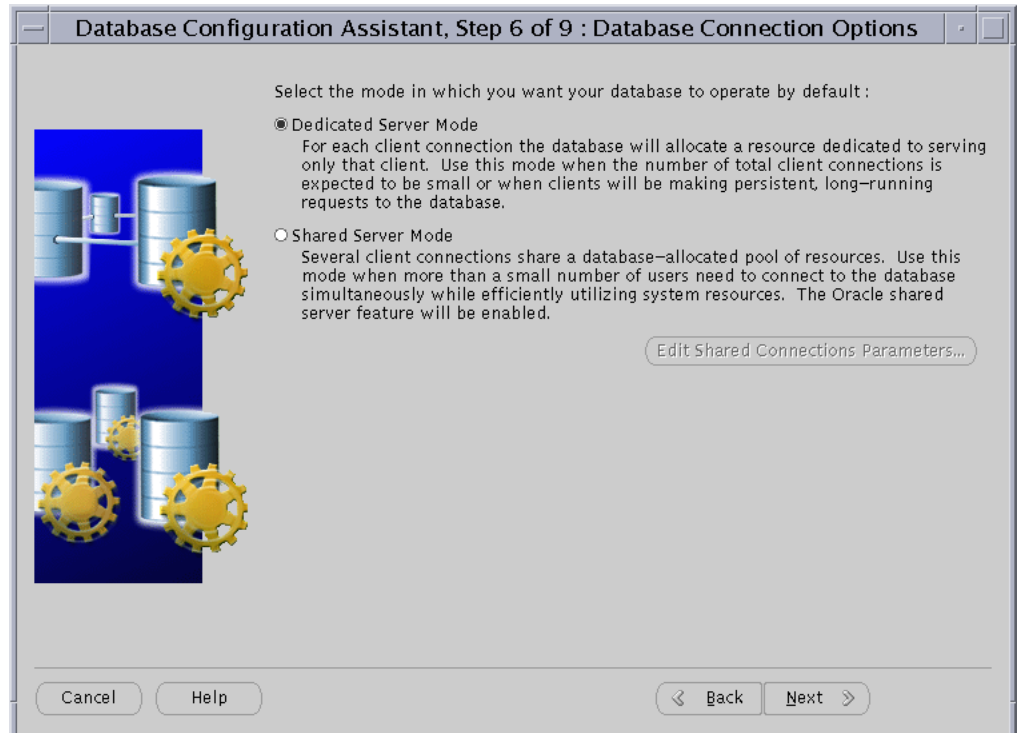
Figure 4-6 Database Configuration Assistant Database Features Page



7. Select the database features that you want to configure in your cluster database and click **Next**.

After you click **Next**, if you have selected the **New Database** template, then the DBCA displays the Database Connection Options page shown in [Figure 4-7](#).

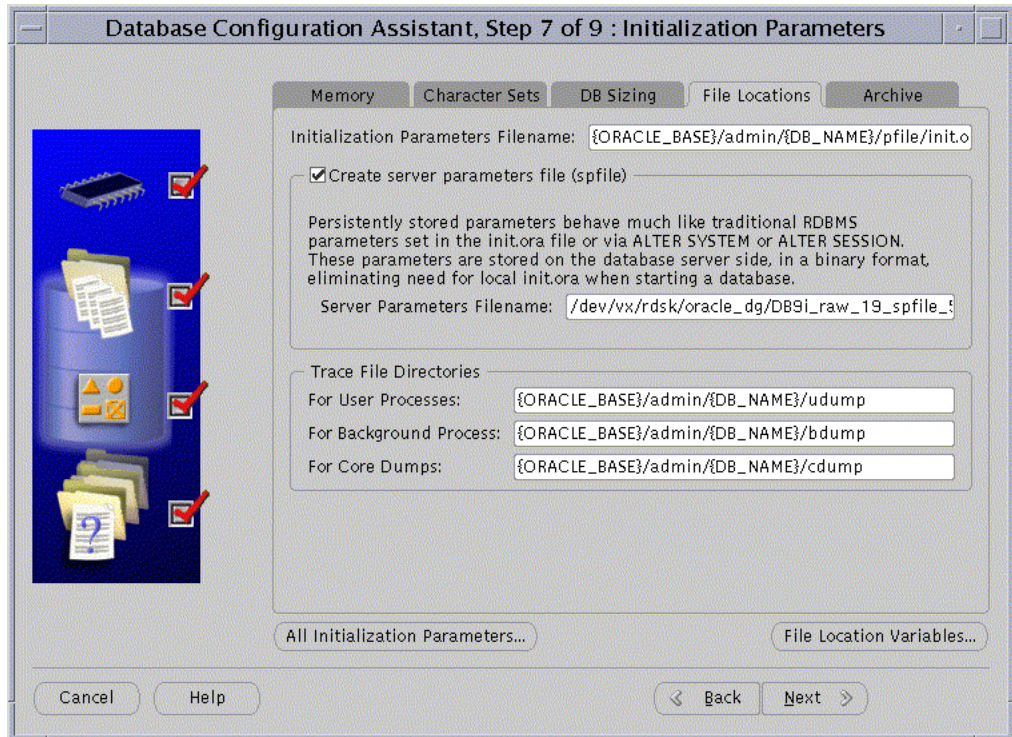
Figure 4-7 Database Configuration Assistant Database Connections Options Page



8. Select the connection mode for your cluster database and click **Next**.

After you click **Next**, the DBCA displays the Initialization Parameters page shown in Figure 4-8.

Figure 4-8 Database Configuration Assistant Initialization Parameters Page



9. Select the **File Locations** tab on the Initialization Parameters page.

If you select **Create server parameter file (spfile)**, then you may need to modify the location for the server parameter file depending on the type of file system you use as described for the following conditions:

- If you use a cluster file system, or if you have a single-node Real Application Clusters environment, then you can place the server parameter file on the file system.
- If you do not use a cluster file system and you are not creating a single-node Real Application Clusters database, then you must enter a raw device name for

the location of the server parameter file in the **Server Parameters Filename** field in the center of the Initialization Parameters page shown in [Figure 4-8](#).

10. Click **All Initialization Parameters** and the DBCA will display the All Initialization Parameters dialog in [Figure 4-9](#). Review the parameter settings that the DBCA will configure in your server parameter file.

Figure 4-9 All Initialization Parameters Dialog

Instance	Name	Value	Included (Y/N)	Category
	parameter_server_fi...	1		Cluster Database
	resource_limit	FALSE		Resource Manager
	max_transaction_...	8		Distributed, Replication and Snapshot
	remote_logn_pas...	exclusive		Security and Auditing
	parallel_instance...			Cluster Database
	timed_os_statistics	0		Diagnostics and Statistics
	dbwr_io_slaves	0		Cache and I/O
	mts_multiple_list...	FALSE		MTS
	max_commit_pro...	700		Cluster Database
	circuits	0		MTS
	nls_territory	AMERICA		NLS
	log_checkpoint_i...	0		Redo Log and Recovery
	transaction_audit...	TRUE		Redo Log and Recovery
	buffer_pool_keep			Cache and I/O
	cluster_database	true		Cluster Database
	cluster_database...	2		Cluster Database
DB9i1	instance_name	DB9i1		Cluster Database
DB9i1	instance_number	1		Cluster Database
DB9i1	thread	1		Cluster Database
DB9i2	instance_name	DB9i2		Cluster Database
DB9i2	instance_number	2		Cluster Database
DB9i2	thread	2		Cluster Database
DB9i1	undo_tablespace	UNDOTBS1		Cluster Database
DB9i2	undo_tablespace	UNDOTBS2		Cluster Database
DB9i1	local_listener	LISTENER_...		Cluster Database
DB9i2	local_listener	LISTENER_...		Cluster Database
	remote_listener	LISTENERS...		Cluster Database

Instance-specific parameter settings for your Real Application Clusters database appear at the bottom of this dialog. The *sid* prefixes for these entries appear in the left-hand column.

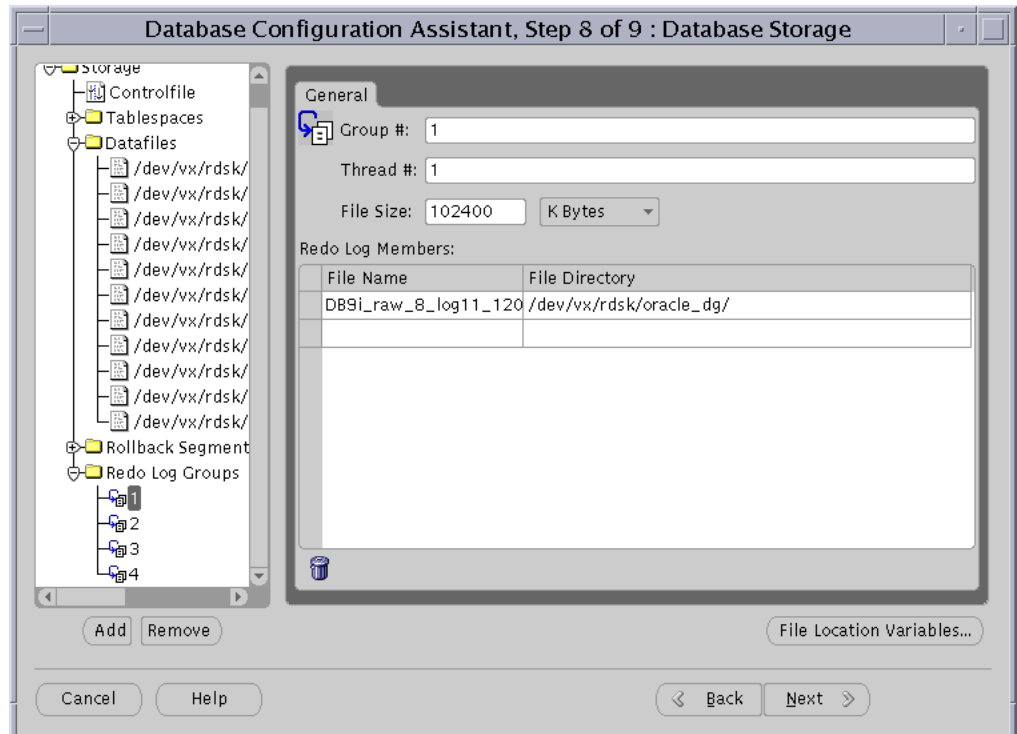
11. To review the instance-specific parameter settings, page downward using the scroll bar on the right-hand side of the dialog.
12. Use the check box in the **Included (Y/N)** column to indicate whether the DBCA should place the parameter setting in your **server parameter file**.

The DBCA only places parameter entries from the All Initialization Parameters dialog into the server parameter file that have check marks in the **Included (Y/N)** column. Also note the following about the All Initialization Parameters dialog:

- You cannot modify the *sid* in the **Instance column**
 - You can alter self-tuning parameters with this dialog, however, setting them to inappropriate values may disable Oracle's self-tuning features
 - You cannot specify instance-specific values for global parameters with the DBCA
13. Complete your entries in the All Initialization Parameters dialog and click **Close**. Make sure your entries for the **File Locations** tab and the other tabs are correct for your Real Application Clusters database and click **Next**.

After you click **Next**, the DBCA displays the Database Storage page similar to the page in [Figure 4–10](#).

Figure 4–10 Database Configuration Assistant Database Storage Page



Use the Database Storage page to enter file names for each tablespace such as SYSTEM, USERS, TEMP, DRSYS, TOOLS, INDX, and so on. The Storage page displays these file names in the **Datafiles** folder.

14. To enter file names for these objects on the Database Storage page, click the **Tablespaces** icon to expand the object tree. Then click the tablespace objects under the tree and replace the default file names with the actual raw device names.

Platform-specific issues for entering file names in the Database Storage page are:

- On UNIX, if you have not set the `DBCA_RAW_CONFIG` environment variable, then the DBCA displays default datafile names. You must override these names

to provide raw device names on this page for each control file, datafile, and redo log group file.

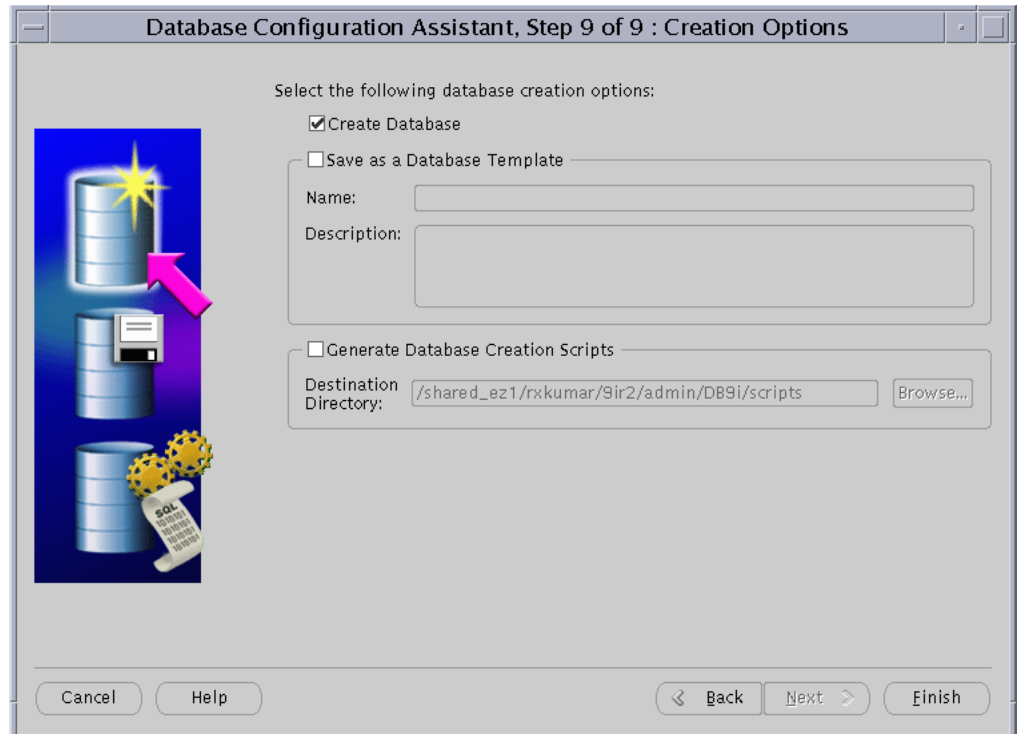
- For Windows NT and Windows 2000, if the default symbolic links exist, then the DBCA replaces the default datafiles with these symbolic link names and displays them in the Storage page. If the symbolic links do not exist, then the DBCA displays the default file system datafiles on the Storage page. In this case, replace these default datafiles with the symbolic link names.

If you select a template that includes datafiles, then the Storage page does not display tablespace information. Instead, the Storage page displays default datafile names that you must replace with raw device filenames. If you are creating a database with a preconfigured database template, then the Storage page does not allow you to change tablespace sizes.

15. To rename default file names if you used a template that includes datafiles, then click the default file name and overwrite it.
16. Click **Next** when you have completed your entries on the Database Storage page.

After you click **Next**, the DBCA displays the Creation Options page shown in [Figure 4-11](#).

Figure 4-11 Database Configuration Assistant Creation Options Page

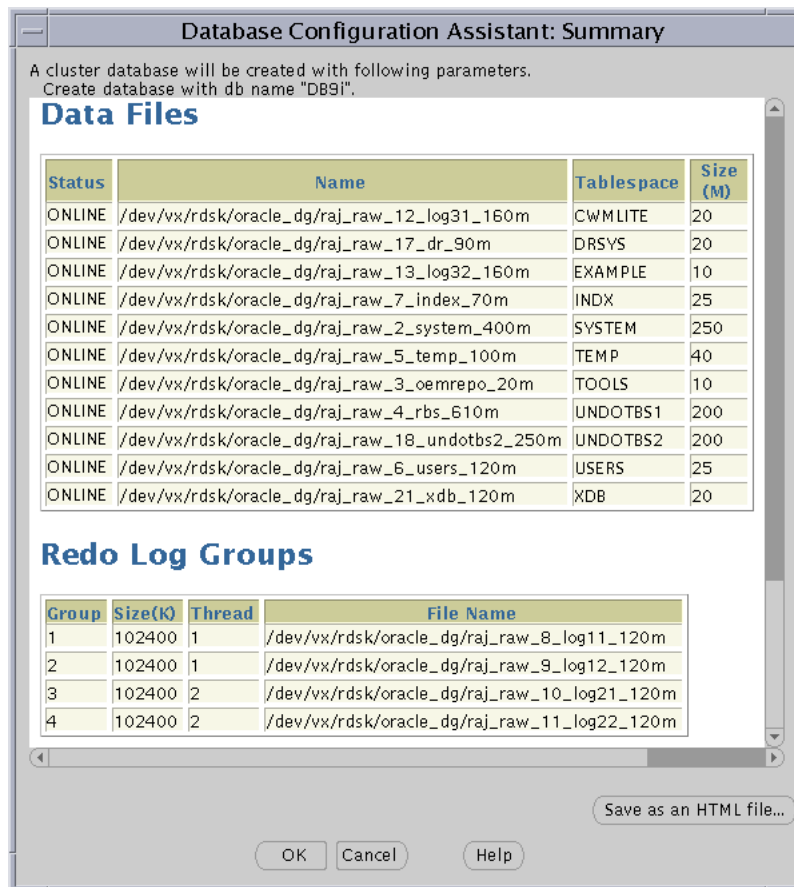


17. Select the database options that you want to use, for example:

- **Create Database**—creates the database
- **Save as a Database Template**—creates a template that records the structure of the database, including user-supplied inputs, initialization parameters, and so on, which you can later use to create a database
- **Generate Database Creation Scripts**—generates database creation scripts

After you click **Finish** on the Creation Options page, the DBCA displays a Summary dialog similar to the dialog in [Figure 4-12](#).

Figure 4-12 Database Configuration Assistant Summary Dialog



18. Review the information on the Summary dialog and click **OK**. To discontinue database creation, click **Cancel**. If you click **OK**, then the DBCA displays database creation progress indicators.

At this point in the database creation process you have:

- Created an operative Real Application Clusters database
- Configured the network for the cluster database

- Started the services if you are on a Windows NT or Windows 2000 platform
- Started the listeners and database instances

Deleting a Database with the Database Configuration Assistant

This section explains how to delete a database with the DBCA. Using the DBCA to delete a database removes a database's initialization parameter files, instances, OFA structure, and Oracle network configuration files. However, the DBCA does not remove datafiles if you placed the files on raw devices or on raw partitions.

To delete a database with the DBCA:

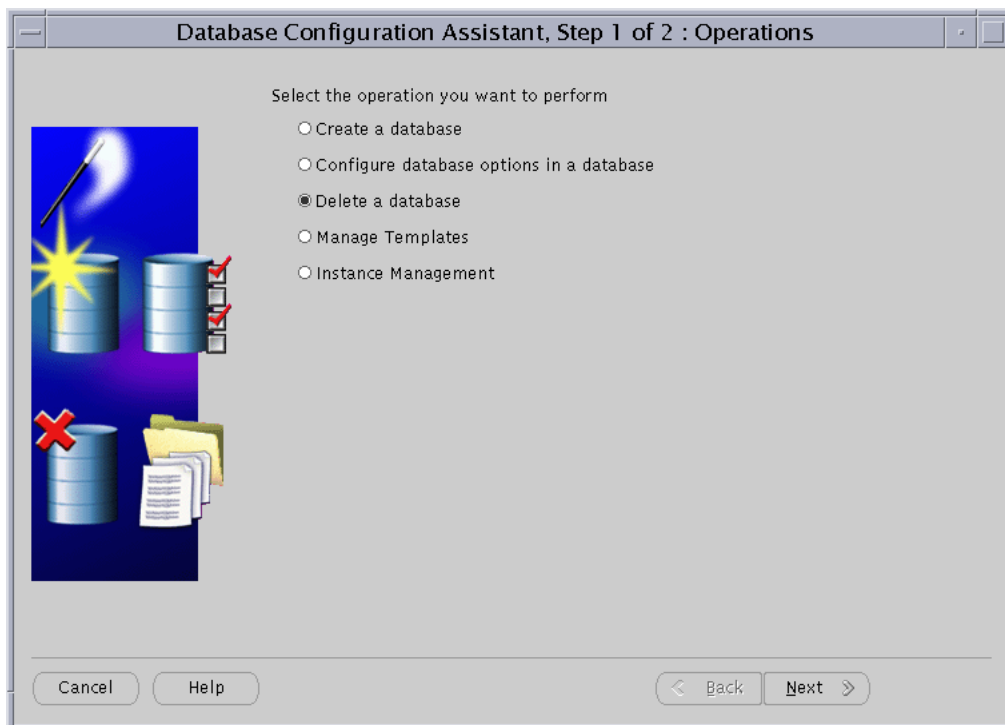
1. Start the DBCA on one of the nodes:
 - On UNIX, execute the `dbca` command from the `$ORACLE_HOME/bin` directory
 - On Windows NT and Windows 2000, choose Start > Programs > Oracle - [HOME_NAME] > Configuration and Migration Tools > Database Configuration Assistant

The DBCA Welcome page appears as shown earlier in this chapter in [Figure 4-1](#).

2. Select **Oracle cluster database** and click **Next**.

After you click **Next**, The DBCA displays the Operations page in [Figure 4-13](#).

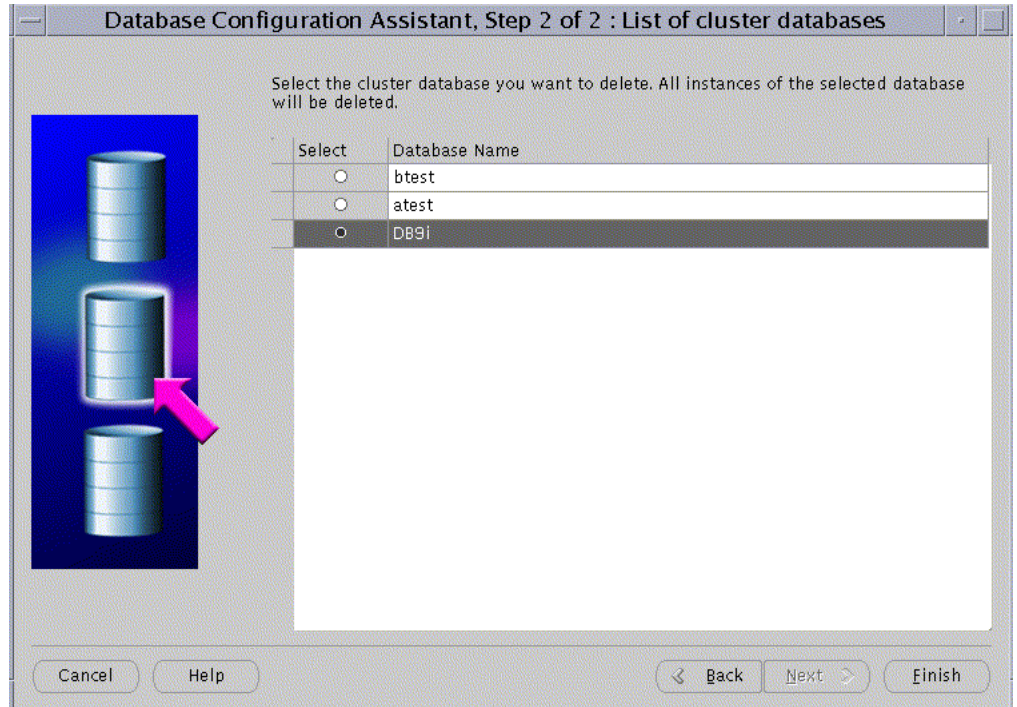
Figure 4-13 Database Configuration Assistant Operations Page



3. Select **Delete a database** and click **Next**.

After you click **Next**, the DBCA displays the List of Cluster Databases page similar to the page in [Figure 4-14](#).

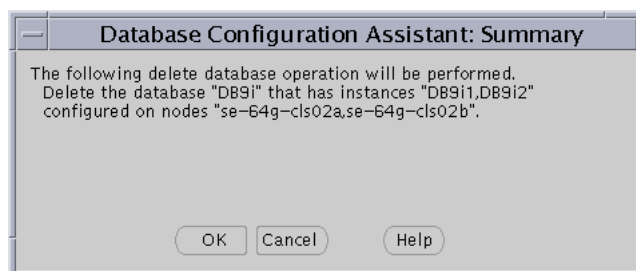
Figure 4-14 Database Configuration Assistant List of Cluster Databases Page



4. If your user ID and password are not operating-system authenticated, then the List of Cluster Databases page displays the user name and password fields. If these fields appear, then enter a user ID and password that has **SYSDBA** privileges.
5. Select the database that you want to delete and click **Finish**.

After you click **Finish**, the DBCA displays a Summary dialog showing the database name and associated instances that the DBCA is going to delete. This Summary dialog, which is similar to the dialog in [Figure 4-15](#), also describes the node or nodes on which the instances reside.

Figure 4-15 Database Configuration Assistant Database Deletion Summary Dialog



6. Click **OK** to begin the deletion of the database and its associated files, services, and environment settings, or click **Cancel** to stop the operation.

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

At this point, you have accomplished the following:

- Stopped the listeners associated with the selected database
- Deleted the selected database from the cluster
- Deleted the selected database's services for Windows NT or Windows 2000
- Deleted the Oracle Net configuration files
- Deleted the Oracle Flexible Architecture directory structure from the cluster

Migrating or Upgrading to Release 2 (9.2)

If the **Oracle Universal Installer (OUI)** detects an earlier version of Oracle, then the Installer prompts you to migrate or upgrade the database to Oracle9i release 2 (9.2).

See Also: The *Oracle9i Database Migration* guide for information about using the Database Upgrade Assistant

Co-Existence of Oracle Versions on the Same Cluster

The co-existence of versions of Oracle database software depends on your operating system platform.

For UNIX operating systems, whether different versions of Oracle can exist on the same cluster is platform-dependent. Refer to your platform-specific Oracle documentation for more information about version co-existence.

For Windows NT and Windows 2000 platforms, as long as your Oracle database software versions are greater than 8.1, they can co-exist on the same cluster when you install them in different locations with different Registry keys. This means that you cannot have different versions of Oracle with version numbers that are previous to release 8.1 on the same cluster. For example, a release 1 (9.0.1) Real Application Clusters database and a release 8.0 Oracle Parallel Server database cannot co-exist on the same cluster.

Co-Existence of Operating System-Dependent Clusterware on Windows Platforms

For Windows NT and Windows 2000, operating system-dependent (OSD) clusterware from Oracle9i release 2 (9.2) can co-exist with previous versions.

Rolling Upgrades

The term rolling upgrades refers to upgrading different databases or different instances of the same database in Oracle9i Real Application Clusters one at a time, without stopping the database. Release 2 (9.2) of Oracle9i Real Application Clusters does not support rolling upgrades.

Multiple Oracle Homes

Oracle9i Real Application Clusters on UNIX, Windows NT, and Windows 2000 platforms supports multiple Oracle homes, just as a single-instance **Oracle9i**

Enterprise Edition database does. The multiple homes feature enables you to install one or more releases on the same machine in multiple Oracle home directories.

Note: Do not move the Oracle binaries from one `ORACLE_HOME` to another because this causes dynamic link time failures. For example on UNIX, the Oracle server requires the shared library `libskgxp9.so` to reside in the `$ORACLE_HOME/lib` directory where Oracle was installed.

Manually Creating Real Application Clusters Databases

This chapter discusses considerations and procedures for manually creating **Real Application Clusters** databases. The topics in this chapter are:

- [Setting CREATE DATABASE Options for Cluster-Enabled Environments](#)
- [Database Objects to Support Clusters](#)
- [Creating a Database Manually](#)

Setting CREATE DATABASE Options for Cluster-Enabled Environments

This section describes the following CREATE DATABASE options specific to Real Application Clusters.

- [Setting MAXINSTANCES](#)
- [Setting MAXLOGFILES and MAXLOGMEMBERS](#)
- [Setting MAXLOGHISTORY](#)
- [Setting MAXDATAFILES](#)

Use this information when writing database creation scripts. A sample database creation script for Real Application Clusters databases appears in the script `clustdb.sql` that resides in the `$ORACLE_HOME/srvm/admin` directory on UNIX or in the `%ORACLE_HOME%\srvm\admin` directory on Windows NT and Windows 2000 platforms.

Setting MAXINSTANCES

The MAXINSTANCES option of CREATE DATABASE limits the number of instances that can access a database concurrently. MAXINSTANCES defaults to the maximum value specific to your operating system.

For Real Application Clusters, set MAXINSTANCES to a value greater than the maximum number of instances you expect to run concurrently. For example, assume you have three instances, A, B, and C. If instance A fails and instance B recovers it, you can start instance C before instance A is fully recovered. In this case, set MAXINSTANCES to 4 or more.

Setting MAXLOGFILES and MAXLOGMEMBERS

The MAXLOGFILES option of CREATE DATABASE specifies the maximum number of redo log groups that can be created for the database. The MAXLOGMEMBERS option specifies the maximum number of members or copies for each group. Set MAXLOGFILES to the maximum number of threads possible, multiplied by the maximum anticipated number of groups for each thread.

Setting MAXLOGHISTORY

The MAXLOGHISTORY clause of the CREATE DATABASE statement specifies the maximum number of archived redo log files that can be recorded in the log history of the control file. The log history is used for automatic media recovery of Real Application Clusters databases.

For Real Application Clusters, set `MAXLOGHISTORY` to a large value, such as 1000. The control files can then only store information about this number of redo log files. When the log history exceeds this limit, Oracle overwrites the oldest entries. The default for `MAXLOGHISTORY` is 0 (zero), which disables log history.

This parameter is useful only if you are using Oracle in ARCHIVELOG mode with Real Application Clusters. Specify the maximum number of archived redo log files for automatic media recovery. Oracle uses this value to determine how much space in the control file to allocate for the names of archived redo log files. The minimum value is 0. The default value is a multiple of the `MAXINSTANCES` value and depends on your operating system. The maximum value is limited only by the maximum size of the control file.

Setting MAXDATAFILES

The `MAXDATAFILES` option is generic, but Real Application Clusters databases tend to have more datafiles and log files than single-instance Oracle databases.

See Also:

- *Oracle9i Real Application Clusters Administration* for more information about redo log groups and members
- *Oracle9i SQL Reference* for complete descriptions of the `CREATE DATABASE` and `ALTER DATABASE SQL` statements

Setting ARCHIVELOG Mode

Create your database using the default of `NOARCHIVE` log mode. This reduces system overhead while you create your database. You can later implement archive logging using the `ALTER DATABASE` statement with the `ARCHIVELOG` option. Refer to "[Setting the Log Mode](#)" on page 5-8 for information on setting the log mode.

Note: You cannot use the `STARTUP` command to change the database archiving mode.

See Also: *Oracle9i Database Administrator's Guide* for more information about archive logging

Changing Values for CREATE DATABASE Options

You can use the `CREATE CONTROLFILE` statement to change the value of the following database parameters for a database:

- `MAXINSTANCES`
- `MAXLOGFILES`
- `MAXLOGMEMBERS`
- `MAXLOGHISTORY`
- `MAXDATAFILES`

See Also: *Oracle9i SQL Reference* for a description of the `CREATE CONTROLFILE` and `ALTER DATABASE BACKUP CONTROLFILE TO TRACE` statements

Database Objects to Support Clusters

To prepare a new database for Real Application Clusters, create and configure the additional database objects as described under the following headings:

- [Creating Additional Undo Tablespaces](#)
- [Configuring the Online Redo Log for Real Application Clusters](#)

Creating Additional Undo Tablespaces

Oracle strongly recommends that you use **automatic undo management**. This feature automatically manages undo space. To use automatic undo management, use the `CREATE DATABASE` statement with the `UNDO TABLESPACE` clause to create an undo tablespace. You can also use the `CREATE UNDO TABLESPACE` statement to create additional undo tablespaces for additional instances.

When you use the `CREATE DATABASE` statement and you have enabled automatic undo management, if you do not specify the `UNDO TABLESPACE` clause, Oracle creates an undo tablespace by default. The name and size of the default file varies depending on your operating system. However, if you are using **raw devices**, this automatically created file will be on your file system where it cannot be shared. In this case, you must specify a name for the undo tablespace.

If you create your database in manual undo management mode, you must first create and bring online one additional rollback segment in the `SYSTEM` tablespace before creating rollback segments in other tablespaces. The instance that creates the

database can create this additional rollback segment and new tablespaces, but it cannot create database objects in non-SYSTEM tablespaces until you bring the additional rollback segment online.

Then you must create at least two rollback segments for each instance of Real Application Clusters. To avoid performance issues, create these rollback segments in a tablespace other than the SYSTEM tablespace, for example, the RBS tablespace.

Note: Do not store these rollback segments in the SYSTEM tablespace.

See Also: *Oracle9i Real Application Clusters Administration* for more information about automatic undo management

Manually Creating Rollback Segments

If you cannot use automatic undo management, then you can manually create rollback segments. Real Application Clusters databases need at least as many rollback segments as the maximum number of concurrent instances plus one; the extra rollback segment is for the SYSTEM rollback segment. An instance cannot start up shared without exclusive access to at least one rollback segment, whether it is public or private.

You can create new rollback segments in any tablespace except for the temporary tablespace. To reduce performance issues between rollback data and table data, partition your rollback segments in a separate tablespace. This facilitates taking tablespaces offline because a tablespace cannot be taken offline if it contains active rollback segments.

In general, make each rollback segment extent the same size by specifying identical values for the storage parameters INITIAL and NEXT. To ensure you have correctly created the rollback segments, examine the data dictionary view DBA_ROLLBACK_SEGS. This view shows each rollback segment's name, segment ID number, and owner (PUBLIC or other).

See Also: *The Oracle9i Database Administrator's Guide* for information about rollback segment performance and for information on the implications of adding rollback segments

Using Private Rollback Segments

If you use manual undo management and manually manage rollback segments, Oracle Corporation recommends that you make the rollback segments private. This enables you to closely control which instances use which rollback segments. To do this follow these steps:

1. Create a rollback segment with the SQL statement using the syntax:

```
CREATE ROLLBACK SEGMENT ... TABLESPACE tablespace_name;
```

2. Use the `ROLLBACK_SEGMENTS` parameter to specify the rollback segment in the **initialization parameter file** by naming it as a value for the parameter. For example, `SID.ROLLBACK_SEGMENTS=[RBS1, RBS2]`. This reserves these two rollback segments for that instance.
3. Use `ALTER ROLLBACK SEGMENT` to bring the rollback segment online. You can also restart the instance to use the reserved rollback segment.

You should specify a particular private rollback segment in either the **server parameter file** with the appropriate instance identifier, or in only one instance-specific initialization parameter file so that the segment is associated with only one instance. If an instance attempts to acquire a public rollback segment that another instance has already acquired, then Oracle generates an error message and prevents the instance from starting up. Private rollback segments stay offline until brought online or until the owning instance restarts and acquires it.

Using Public Rollback Segments

Any instance can create public rollback segments. Once created, public rollback segments are available for any instance. When an instance uses a rollback segment, the instance uses the rollback segment exclusively until the instance shuts down. When the instance shuts down, the instance releases the rollback segment for use by other instances.

Use the SQL statement `CREATE PUBLIC ROLLBACK SEGMENT` to create public rollback segments. Public rollback segments are owned as `PUBLIC` in the data dictionary view `DBA_ROLLBACK_SEGS`. If you do not assign a rollback segment to an instance by setting a value for the `ROLLBACK_SEGMENTS` parameter, the instance uses public rollback segments. The procedures you use to create and manage rollback segments are the same regardless of whether you have enabled or disabled Real Application Clusters.

Typically, the parameter file does not specify public rollback segments because they are by default available to any instance needing them. However, if another instance is not already using a particular public rollback segment, you can assign the

rollback segment to the instance by declaring a value in the `ROLLBACK_SEGMENTS` parameter for that instance.

An instance brings a public rollback segment online when the instance acquires the rollback segment at startup. However, starting an instance that uses public rollback segments does not ensure that the instance uses a particular public rollback segment. The exception to this is when the instance acquires all available public rollback segments.

If you need to keep a public rollback segment offline and do not want to drop it and re-create it, you must prevent other instances that require public rollback segments from starting up.

See Also: *Oracle9i Database Administrator's Guide* for more information about rollback segments

Configuring the Online Redo Log for Real Application Clusters

When running Real Application Clusters, two or more instances concurrently access a single database and each instance must have its own thread of redo. This section explains how to configure these online redo threads for clusters.

Each database instance has its own *thread* of online redo, consisting of its own online redo log groups. Oracle Corporation recommends that you create at least two members for each redo log group to prevent data loss. Create each thread with at least two redo log groups and enable each thread so the instance can use it.

For improved performance and to minimize the overhead of software mirroring, or *multiplexing*, put the members of each redo log group on separate physical disks or on separate disk arrays. The `CREATE DATABASE` statement creates thread number 1 as a public thread and enables it automatically. Use the `ALTER DATABASE` statement to create and enable subsequent threads.

Creating Threads

Threads can be either public or private. The `THREAD` initialization parameter assigns a unique thread number to an instance. If you set `THREAD` to zero, which is the default, the instance acquires a public thread.

The `CREATE DATABASE` statement creates thread number 1 as a public thread and enables it automatically. Subsequent threads must be created and enabled with the `ALTER DATABASE` statement. For example, the following statements create and enable thread 2 with two groups of three members each.

```
ALTER DATABASE ADD LOGFILE THREAD 2
```

```
GROUP 3 (disk1_file4, disk2_file4, disk3_file4) SIZE 100M REUSE,  
GROUP 4 (disk1_file5, disk2_file5, disk3_file5) SIZE 100M REUSE;  
ALTER DATABASE ENABLE PUBLIC THREAD 2;
```

If you do not specify the `THREAD` parameter in your initialization file, you must specify the `THREAD` clause when creating new redo log groups. If you specify the `THREAD` parameter, you can omit the `THREAD` clause when creating new redo log groups and the newly created redo log groups will be assigned to the thread of the instance that you used to create them.

See Also: *Oracle9i Real Application Clusters Administration* for more information about threads of redo

Disabling Threads

Disable a public or private thread with the `ALTER DATABASE DISABLE THREAD` statement. You cannot disable a thread if an instance using the thread has the database mounted. To change a thread from public to private, or vice versa, disable the thread and then enable it again. An instance cannot disable its own thread. The database must be open when you disable or enable a thread.

When you disable a thread with the database in `ARCHIVELOG` mode, Oracle marks its current redo log file as needing to be archived. If you want to drop that file, you might need to first archive it manually.

An error or failure while a thread is being enabled can result in a thread that has a current set of log files but is not enabled. You cannot drop or archive these log files. In this case, disable the thread, even though it is already disabled, then re-enable it.

Setting the Log Mode

You typically set the redo log mode, `ARCHIVELOG` or `NOARCHIVELOG`, when you create your database. You can later change the archive mode using the `ALTER DATABASE` statement. When archiving is enabled, online redo log files cannot be reused until they are archived.

The redo log mode is associated with the database rather than with individual instances. If the redo log is being used in `ARCHIVELOG` mode, for most purposes all instances should use the same archiving method, either automatic or manual.

To switch archiving modes:

1. Set the `CLUSTER_DATABASE` parameter to `false` in the parameter file.
2. Mount the database in exclusive mode.

3. Set the `LOG_ARCHIVE_START` parameter to `true`.
4. Set the `LOG_ARCHIVE_FORMAT` and `LOG_ARCHIVE_DEST_n` parameters as needed.
5. Execute the `ALTER DATABASE` statement with either the `ARCHIVELOG` or the `NOARCHIVELOG` clause.
6. Shutdown the database and then restart it with the `CLUSTER_DATABASE` parameter set to `true`.

Changing the Redo Log

You can change the configuration of the redo log, such as adding, dropping, or renaming a log file or log file member, while the database is mounted with Real Application Clusters either enabled or disabled. The only restriction is that you cannot drop or rename a log file or log file member currently in use by any thread. Moreover, you cannot drop a log file if that would reduce the number of log groups to less than two for the thread it is in.

Any instance can add or rename redo log files, or members, of any group for any other instance. As long as there are more than two groups for an instance, a redo log group can be dropped from that instance by any other instance. Changes to redo log files and log members take effect on the next log switch.

See Also: *Oracle9i Real Application Clusters Administration* for more information about archiving redo log files

Creating a Database Manually

Create your database manually if you already have scripts, or if you have database requirements that differ greatly from the types of databases that the DBCA creates as described in [Chapter 4](#). The two major steps you must perform to manually create a database are:

- [Install Oracle Products](#)
- [Manually Create the Database](#)
- [Creating Real Application Clusters Data Dictionary Views with CATCLUST.SQL](#)

Install Oracle Products

Perform the following tasks before manually creating a Real Application Clusters database:

1. Run the **Oracle Universal Installer (OUI)** as specified in your platform-specific documentation.
2. At the Installation Types screen select **Software Only** and click **Continue**. Your installation should proceed automatically without installing a Real Application Clusters database.

Refer to the following section for procedures on manual database creation.

Manually Create the Database

Perform the following tasks to manually create a new database:

[Task 1: Back Up Existing Databases](#)

[Task 2: Specify the Database and Instance Parameter Settings](#)

[Task 3: Create the Real Application Clusters Configuration with SRVCTL](#)

[Task 4: Configure the oratab File on UNIX](#)

[Task 5: Set ORACLE_SID for Each Node](#)

[Task 6: Create the Server Parameter File](#)

[Task 7: Create the Password Files](#)

[Task 8: Prepare a CREATE DATABASE Script for the Cluster Database](#)

[Task 9: Create the Database](#)

[Task 10: Back Up the Database](#)

[Task 11: Configure Oracle Net on All Nodes](#)

Review all the steps in this chapter before performing them.

Task 1: Back Up Existing Databases

Oracle strongly recommends that you make complete backups of all existing databases before creating a new database in case database creation accidentally affects existing files. Backups should include parameter files, database files, redo log files, control files, and network configuration files.

See Also: *The Oracle9i User-Managed Backup and Recovery Guide*

Task 2: Specify the Database and Instance Parameter Settings

In Real Application Clusters, each **node** typically has one instance. The instances collectively form a Real Application Clusters database. Being aware of database- and instance-level information enables you to more easily complete Tasks 2 through 12.

Note: You can configure multiple instances on one node, but this may reduce scalability.

To determine database- and instance-level information:

1. Determine the settings for your database for the items in the left-hand column of [Table 5-1](#):

Table 5-1 *Component Settings for Manual Database Creation*

Component	Description
Database Name	The name of your database.
Database Domain	The domain name of your database.
Global Database Name	A name that comprises the database name and database domain.
SID Prefix	A prefix for the Oracle system identifier (sid) . The instance's thread number , or number of the redo thread assigned to the instance, is appended to the SID prefix to create the SID for the node's instance.

2. Determine the settings for the items in [Table 5-2](#). For example, [Table 5-2](#) shows the settings for a database named db:

Table 5-2 *Database Name, Domain, Global Name, and SID Prefix Example Settings*

Database Name	Database Domain	Global Database Name	SID Prefix
db	us.acme.com	db.us.acme.com	db

- Determine the settings for the items in the left-hand column of [Table 5-3](#) for each node:

Table 5-3 Component Settings for Node and Host Name and Thread ID

Component	Description
Node name	<p>The node name defined by the Cluster Manager (CM) software</p> <p>Use the command <code>lsnodes -l -n</code> to obtain the computer's node name.</p> <p><code>lsnodes</code> is located in the <code>\$ORACLE_HOME/bin</code> directory on UNIX and <code>%ORACLE_HOME%\bin</code> directory on Windows NT and Windows 2000.</p>
Host name	<p>The host name of the computer. The host name may be the same name as the node name.</p> <p>On UNIX and Windows NT and Windows 2000, use the command <code>hostname</code> to obtain the host name for the computer.</p>
Thread ID	<p>Each instance requires a unique thread number. The thread ID is appended to the SID prefix to create the SID for the instance on the node. Threads are usually numbered sequentially beginning with 1.</p>

- Determine the settings for the items in the column headers in [Table 5-4](#), which for example, shows the nodes named `node1` and `node2`:

Table 5-4 Host and Node Name, Thread ID and SID

Node Name	Host Name	Thread ID	SID
<code>node1</code>	<code>node1</code>	1	<code>db1</code>
<code>node2</code>	<code>node2</code>	2	<code>db2</code>

Task 3: Create the Real Application Clusters Configuration with SRVCTL

If this is the first Oracle9i database created on this **cluster database**, then you must initialize the clusterwide SRVM configuration. Do this by executing the following command:

```
srvconfig -init
```

The first time you use the **SRVCTL Utility** to create the configuration, start the **Global Services Daemon (GSD)** on each node with the `gsdctl start` command so that SRVCTL can access your cluster's configuration information. Then execute the `srvctl add` command so that Real Application Clusters knows what instances belong to your cluster using the following syntax.

If you are using the server parameter file, then execute the following command:

```
srvctl add db -d db_name -m db_domain -o oracle_home -s spfile_name
```

If your database does not have a domain name, then do not specify the `-m` option. If you do not use the server parameter file, then do not specify the `-s` option. Then for each instance enter the command:

```
srvctl add instance -d db_name -i sid -n node
```

Task 4: Configure the oratab File on UNIX

To use **Oracle Enterprise Manager**, manually create an entry in the `oratab` file on each node. This entry identifies the database. Oracle Enterprise Manager uses the information in this file during **service discovery** to determine the database name and the `ORACLE_HOME` from which the database runs.

The `oratab` file is stored in `/etc/oratab` or `/var/opt/oracle/oratab`, depending on your operating system. The syntax for this entry is as follows where `db_name` is the database name given to your database, `$ORACLE_HOME` is the directory path to the database, and `N` indicates that the database should not be started at restart time:

```
db_name:$ORACLE_HOME:N
```

Use the database name and Oracle home you specified in "[Task 2: Specify the Database and Instance Parameter Settings](#)" on page 5-11.

A sample entry follows for a database named `db`:

```
db:/private/oracle/db:N
```

Task 5: Set ORACLE_SID for Each Node

The SID must be defined for each node's instance in the **cluster database**, and the value you set for `ORACLE_SID` must be unique for each instance. To simplify administration, Oracle Corporation recommends that you use SIDs that consist of the database name as the common base and the number of the thread assigned to the instance that you specified in "[Task 2: Specify the Database and Instance](#)

[Parameter Settings](#)". For example, if `db` is the database name, then the first instance in the cluster has a SID of `db1` and the second instance has a SID of `db2`. The SID specification is operating system-specific as described under the following headings:

- [UNIX](#)
- [Windows NT and Windows 2000](#)

UNIX On UNIX, set the `ORACLE_SID` environment variable.

See Also: *Oracle9i Administrator's Reference* for your UNIX operating system for further information about setting this environment variable

Windows NT and Windows 2000 On Windows NT and Windows 2000, create an `ORACLE_SID` Registry key under the following Registry key. Then set the value of your instance SID in the `ORACLE_SID` registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\[HOMEID]
```

See Also: *Oracle9i Database Getting Started for Windows* for further information about this Registry value

After creating the SIDs, create an `OracleServicesid` service. You can use this service to start or stop an instance from the Control Panel. To create `OracleServicesid`:

1. On *each* node, use the `CRTSRV` batch file to create a unique service corresponding to the instance of the node.

```
C:\%ORACLE_HOME%\bin\ crtsrv.bat sid
```

For example, to create a service for a SID of `db1`, `OracleServicedb1`, enter the following:

```
C:\%ORACLE_HOME%\bin\ crtsrv.bat db1
```

2. Verify `OracleServicesid` exists by entering:

```
C:\net start
```

Note: The Service must already have a status of `Started` to appear in the list that Oracle displays in response to this command.

Task 6: Create the Server Parameter File

When an Oracle instance starts, it refers to the parameter file for configuration information. Oracle Corporation recommends that you use a single server parameter file to designate both global and instance-specific settings. Using this type of parameter file greatly simplifies parameter administration.

Name the server parameter file **initdbname.ora**. To designate instance-specific settings in this file, use the SID prefix and place these entries after the generic, global entries. Specify instance-specific settings using the *instance_name.parameter_name=value* syntax.

You can also make parameter files for the database you are about to create by copying the initialization parameter file located in the `$ORACLE_HOME/srvm/admin` directory on UNIX or by using the file in the `%ORACLE_HOME%\srvm\admin` directory on Windows NT and Windows 2000. Rename this file and edit and customize it for your database.

Edit the following parameters in the new *initdb_name.ora* parameter file:

- REMOTE_LOGIN_PASSWORD
- BACKGROUND_DUMP_DEST
- CONTROL_FILES
- DB_DOMAIN
- DB_NAME
- DISPATCHERS
- SERVICE_NAMES
- USER_DUMP_DEST
- UNDO_MANAGEMENT

Edit the following instance-specific parameters in the file:

- INSTANCE_NAME
- INSTANCE_NUMBER
- UNDO_TABLESPACE or ROLLBACK_SEGMENTS

- THREAD
- LOCAL_LISTENER

Examples of instance-specific settings in the server parameter file are:

```
db1.instance_name=db1
db1.instance_number=1
db2.instance_name=db2
db2.instance_number=2
```

3. Ensure that the parameter `REMOTE_LOGIN_PASSWORDFILE` is set to `EXCLUSIVE`.

See Also: ["Initialization Parameter Files"](#) on page 8-7 for further information about initialization parameter files and the parameters to set

Task 7: Create the Password Files

Use the Password Utility `ORAPWD` to create password files. `ORAPWD` is automatically installed with the Oracle9i utilities. Password files are located in the `$ORACLE_HOME/dbs` directory on UNIX and in the `%ORACLE_HOME%\database` directory on Windows NT and Windows 2000. They are named `orapwsid` on UNIX and `pwdsid.ora` on Windows NT and Windows 2000, where `sid` identifies the database instance you specified in ["Task 2: Specify the Database and Instance Parameter Settings"](#) on page 5-11.

To create a password file on each node:

1. Use `ORAPWD` to create the password file.
 - On UNIX, run `orapwd` from `$ORACLE_HOME/bin` with the following syntax:

```
orapwd file=$ORACLE_HOME/dbs/orapw$ORACLE_sid password=password
```

- On Windows NT and Windows 2000, run `orapwd` from `%ORACLE_HOME%\bin` with the following syntax:

```
ORAPWD file=%ORACLE_HOME%\database\pwdsid.ora password=password
```

`FILE` specifies the password file name and `PASSWORD` sets the password for the `SYS` account.

Task 8: Prepare a CREATE DATABASE Script for the Cluster Database

Prepare a `CREATE DATABASE` script on one of the nodes by using the `clustdb.sql` sample script, located in the `$ORACLE_HOME/srvm/admin` directory on UNIX or in the `%ORACLE_HOME%\srvm\admin` directory on Windows NT and Windows 2000. The sample script is for a two-node cluster. If you use the sample script, edit the following:

1. Set `PFILE` to point to the location of the `initdb_name.ora` file.

Note: You must use the `PFILE` parameter to derive the initial parameter settings from which you create the server parameter file.

2. Modify `oracle` in the `CONNECT SYS/oracle AS SYSDBA` line to use the password you created in ["Task 7: Create the Password Files"](#) on page 5-16. You must also be a member of the **SYSDBA** group.
3. Modify the location of the data dictionary scripts, `$ORACLE_HOME/rdbms/admin` on UNIX and `%ORACLE_HOME%\rdms\admin` on Windows NT and Windows 2000, to reflect the Oracle home you specified in ["Task 2: Specify the Database and Instance Parameter Settings"](#) on page 5-11.
4. Modify the log file and datafile names with the file names or symbolic link names you created in ["Configuring Shared Disk Subsystems for Real Application Clusters"](#) on page 2-2.
5. Modify the log file and datafile sizes.
6. If you are not using automatic undo management, then create enough private rollback segments for the number of concurrent users for each transaction. Oracle requires at least two rollback segments for each instance. With the exception of the `SYSTEM` rollback segment, instances cannot share public rollback segments. An instance explicitly acquires private rollback segments when it opens a database.

Task 9: Create the Database

To create the new database, run the `CREATE DATABASE SQL` script (`clustdb.sql`) from the SQL*Plus prompt:

```
@path/clustdb.sql;
```

The location of the `clustdb.sql` script is `$ORACLE_HOME/srvm/admin` on UNIX and `%ORACLE_HOME%\srvm\admin` on Windows NT and Windows 2000 platforms.

When you execute this script, Oracle creates the following:

- Control files for the database
- Datafiles for the database
- Redo log files for the database
- Data dictionary
- SYSTEM tablespace and the SYSTEM rollback segment
- Users SYS and SYSTEM

Then Oracle mounts and opens the local database instance for use.

Task 10: Back Up the Database

Make a full backup of the database to ensure you have a complete set of files from which to recover in case of media failure.

See Also: *Oracle9i User-Managed Backup and Recovery Guide*

Task 11: Configure Oracle Net on All Nodes

Configure the [listener.ora](#), [sqlnet.ora](#), and [tnsnames.ora](#) files as described in [Table 5-5](#):

Table 5-5 Oracle Net Configuration File Requirements

Configuration File	Description	Configuration Requirements
<code>listener.ora</code>	Includes addresses of each network listener on a server, the SIDs of the databases for which they listen, and various control parameters used by the listener.	<p>The <code>listener.ora</code> file on each node must be configured with:</p> <ul style="list-style-type: none"> ▪ Listener name ▪ A TCP/IP address for Oracle Enterprise Manager ▪ An entry for the SID of the instance in the <code>sid_LIST_listener_name</code> section <p>See Also:</p> <ul style="list-style-type: none"> ▪ The "Configuring and Administering the Listener" chapter in the <i>Oracle9i Net Services Administrator's Guide</i> for configuration procedures
<code>tnsnames.ora</code>	Includes a list of network descriptions of each service name , called net service names .	<p>See Also:</p> <ul style="list-style-type: none"> ▪ "Net Service Names (<code>tnsnames.ora</code> File)" on page 8-12 for net service name requirements ▪ The "Configuring Naming Methods" chapter in the <i>Oracle9i Net Services Administrator's Guide</i> for configuration procedures
<code>sqlnet.ora</code>	Includes the names resolution method.	<p>Because the net service names are specified in <code>tnsnames.ora</code> files, the <code>sqlnet.ora</code> file must specify that the <code>tnsnames.ora</code> file be used when resolving a net service name.</p> <p>See Also: "Profile (<code>sqlnet.ora</code> File)" on page 8-16 for a sample configuration</p>

See Also: *Oracle9i Net Services Administrator's Guide* for information about creating these files

Creating Real Application Clusters Data Dictionary Views with CATCLUST.SQL

Because you manually created your Real Application Clusters database, you must run the `CATCLUST.SQL` script to create Real Application Clusters-related views and tables. You must have `SYSDBA` privileges to run this script.

See Also: *Oracle9i Database Reference* for more information on dynamic performance views

Configuring the Server Parameter File in Real Application Clusters Environments

This chapter describes server parameter file placement and configuration in **Real Application Clusters** environments. The topics in this chapter are:

- [Parameter Files and Real Application Clusters](#)
- [Using The Server Parameter File in Real Application Clusters](#)
- [Parameter File Search Order](#)
- [Migrating to the Server Parameter File in Real Application Clusters Environments](#)
- [Server Parameter File Errors](#)
- [Backing Up the Server Parameter File](#)

See Also: *Oracle9i Real Application Clusters Administration* for more information about parameters in Real Application Clusters environments and *Oracle9i Real Application Clusters Deployment and Performance* for a discussion of parallel execution-related parameters in Real Application Clusters data warehouse environments

Parameter Files and Real Application Clusters

Oracle uses parameter settings in parameter files to determine how to control various database resources. You can use two types of files for parameter administration: the **server parameter file** or one or more traditional client-side parameter files.

Oracle Corporation recommends that you administer parameters using the server parameter file. If you use the traditional client-side parameter files, parameter changes that Oracle makes for self-tuning are not preserved after shutdown.

See Also: *Oracle9i Real Application Clusters Administration* for more information on using client-side parameter files in Real Application Clusters

Using The Server Parameter File in Real Application Clusters

By default, Oracle creates the server parameter file based on one `SPFILE`. You can only change parameter settings in the server parameter file using **Oracle Enterprise Manager** or `ALTER SYSTEM SET SQL` statements; the server parameter file is a binary file that you cannot directly edit.

Note: Oracle Corporation recommends that you avoid modifying the values for self-tuning parameters; overriding these settings can adversely affect performance.

If you are migrating from a previous Oracle release, create and configure the server parameter file for Real Application Clusters using the procedures described in the following section.

Location of The Server Parameter File

The default location of the server parameter file when the database creates it from `PFILEs` is platform-dependent.

The default location of the server parameter file on UNIX is:

```
$ORACLE_HOME/dbs/spfile.$ORACLE_SID.ora
```


The default location of the server parameter file on Windows NT and Windows 2000 is:

```
%ORACLE_HOME%\database\SPFILE%ORACLE_SID%.ORA
```

The default location of the server parameter file is inappropriate for Real Application Clusters databases because all instances must use the same server parameter file.

Instead, for UNIX platforms Oracle Corporation recommends that you use a PFILE in:

```
$ORACLE_HOME/dbs/init.$ORACLE_SID.ora
```

For Windows NT and Windows 2000 platforms Oracle Corporation recommends that you use a PFILE in:

```
%ORACLE_HOME%\database\init%ORACLE_SID%.ora
```

This file is for each instance and it references a single, shared initialization parameter file. The file must contain the following entry for UNIX platforms:

```
SPFILE= '/dev/vx/rdisk/oracle_dg/db_spfile'
```

The file must contain the following entry for Windows NT and Windows 2000 platforms:

```
SPFILE='db_spfile1'
```

However, if you use a cluster file system, then use an alternate file location of:

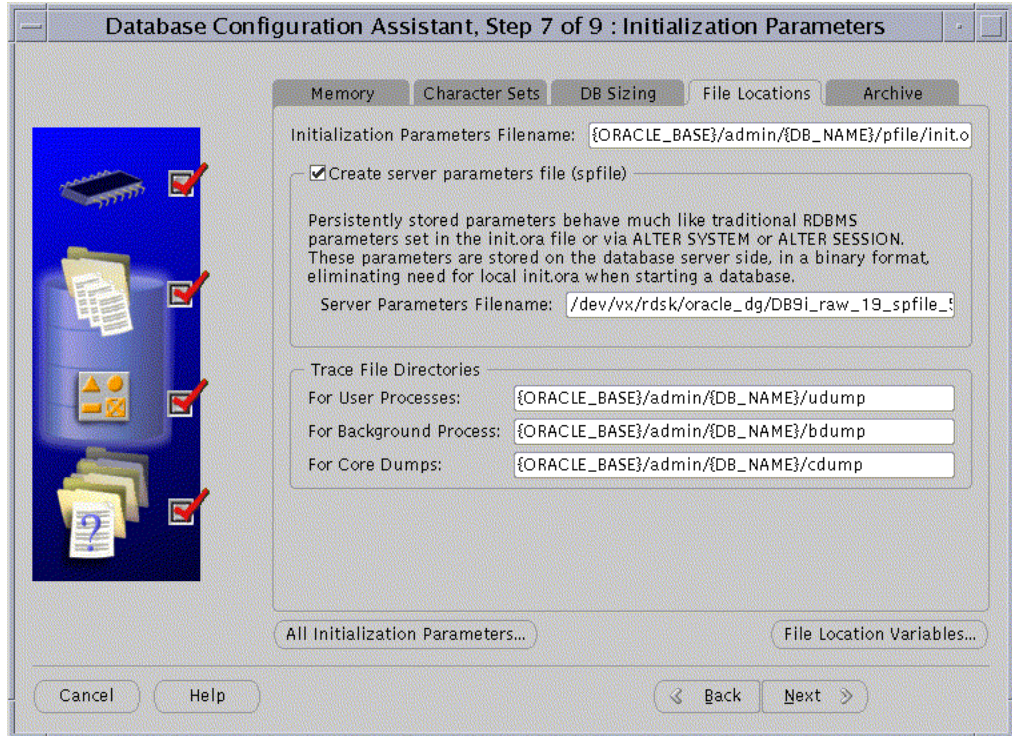
```
SPFILE= $ORACLE_HOME/dbs/spfile.ora
```

You must use the same value of SPFILE so that all instances use the same server parameter file at startup.

To use the DBCA to create your database and to use the server parameter file, on the Initialization Parameters page select the **Create server parameter file (spfile)** box under the File Locations tab. Then enter either a shared file system filename or

the **raw device** path name in the **Server Parameters Filename** field as shown in **Figure 6-1**.

Figure 6-1 File Locations Tab For Initialization Parameters



Note: When you use the DBCA to create the server parameter file, the PFILE file name is `$ORACLE_HOME/dbs/init$ORACLE_SID.ora` on UNIX and `%ORACLE_HOME%\database\init%ORACLE_SID%.ora` on Windows NT and Windows 2000. These are the default PFILE names.

Parameter File Search Order

Oracle searches for your initialization parameter file in a particular order depending on your platform. On UNIX platforms, Oracle examines directories in the following order:

1. `$ORACLE_HOME/dbs/spfilesid.ora`
2. `$ORACLE_HOME/dbs/spfile.ora`
3. `$ORACLE_HOME/dbs/initSID.ora`

The search order on Windows NT and Windows 2000 platforms is:

1. `%ORACLE_HOME%\database\spfilesid.ora`
2. `%ORACLE_HOME%\database\spfile.ora`
3. `%ORACLE_HOME%\database\initSID.ora`

Migrating to the Server Parameter File in Real Application Clusters Environments

To migrate to the server parameter file, create and edit the server parameter file using the procedures described in this section.

Server Parameter File Placement in Real Application Clusters

Put the server parameter file on a raw device that is at least 5MB in size. For single-**node** cluster-enabled configurations, or if you are using a cluster file system, place the server parameter file on a file system.

Procedures for Migrating to the Server Parameter File

Migrate to the server parameter file by completing the following procedures:

Note: The following procedures show examples for UNIX only.

1. Combine the **initialization parameter files** for all instances into a single `initdbname.ora` file by copying all shared `IFILE` contents as *is*. All parameters defined in your `IFILE` parameter files are global. Therefore, create them as "parameter=value" without **sid** prefixes.

2. Copy all instance-specific parameter definitions from `initsid.ora` files using the following syntax where `sid` is the `sid` of the instance:

```
sid.parameter=value
```

3. Create the server parameter file using the `CREATE SPFILE` statement. For example:

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

This statement reads your combined `initdbname.ora` file that you created by merging your `IFILES`. Then it transfers the settings for the parameters from the merged file into your server parameter file.

4. Oracle Corporation recommends that you use the server parameter file by executing the `STARTUP` command as in this example:

```
STARTUP PFILE=$ORACLE_HOME/dbs/initsid.ora
```

Where the file `initsid.ora` contains the entry:

```
SPFILE='/dev/vx/rdisk/oracle_dg/db_spfile'
```

If you use this `STARTUP` command syntax, Oracle uses the server parameter file entry specified in `initsid.ora`.

Note: The release 8.1 default of using a client-side `PFILE` to startup a database instance is no longer supported.

Server Parameter File Errors

Oracle reports errors that occur during the server parameter file creation or while reading the file during startup. If an error occurs during a parameter update, Oracle records the error in your `ALERT.LOG` file and ignores subsequent parameter updates to the file. If this happens, you can do any of the following:

- Shutdown the instance, recover the server parameter file, and restart the instance.
- Allow the instance to continue running without regard for subsequent parameter updates.

Oracle displays errors for parameter changes that you attempt when you incorrectly use the `ALTER SYSTEM SET` statement. Oracle does this when an error occurs while reading from or writing to the server parameter file.

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` statement. For example:

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

You can also recover by starting up an instance using a client-side initialization parameter file. Then re-create the server parameter file using the `CREATE SPFILE` statement. You cannot use **RMAN (Recovery Manager)** to create backups of the server parameter file.

See Also: *Oracle9i SQL Reference* for more information about the `CREATE SPFILE` statement

Part II

Oracle Enterprise Manager Installation Highlights for Real Application Clusters

Part II describes issues for installing Oracle Enterprise Manager and its Server Management (SRVM) and Performance Manager subcomponents in Real Application Clusters environments. The chapter in Part II is:

- [Chapter 7, "Configuration Highlights for Using Enterprise Manager in Real Application Clusters Environments"](#)

Configuration Highlights for Using Enterprise Manager in Real Application Clusters Environments

This chapter describes the highlights of installing and configuring **Oracle Enterprise Manager** for use in Real Application Clusters environments. The topics in this chapter include:

- [Server Management Architecture](#)
- [Server Management Requirements](#)
- [Understanding the Oracle Enterprise Manager Setup](#)
- [Configuring Enterprise Manager for Real Application Clusters Management](#)
- [Using Oracle Performance Manager for Real Application Clusters Management](#)

Server Management Architecture

Oracle Enterprise Manager (EM) supports the administration of cluster databases by using **Server Management (SRVM)** to perform Real Application Clusters-specific tasks. SRVM includes all the components of EM, such as the Intelligent Agent, the Server Control Utility (SRVCTL), and the Global Services Daemon. Another performance monitoring component of Enterprise Manager, called **Oracle Performance Manager**, further enhances SRVM by enabling you to monitor cluster database performance using the global GV\$ view tables.

EM provides tools to manage, monitor, and administer even the most complex network of databases from a single point known as the Enterprise Manager Console. You can use EM with single-instance Oracle databases as well as with Real Application Clusters databases. Within the context of EM, SRVM includes the instance management utilities shown in [Table 7-1](#):

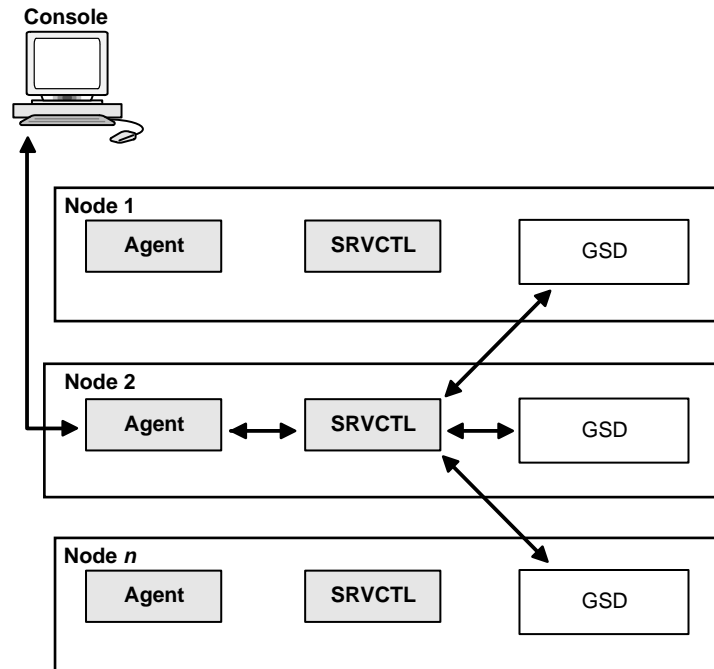
Table 7-1 SRVM Instance Management Utilities

Component	Description
Server Control (SRVCTL)	<p>The SRVCTL Utility serves as a single point of control between the Oracle Intelligent Agent and each node.</p> <p>SRVCTL uses RMI to communicate with the Global Services Daemon on other nodes. SRVCTL is installed on the nodes.</p>
Global Services Daemon (GSD)	<p>Global Services Daemon (GSD) receives requests from a SRVCTL command to execute administrative tasks, such as startup or shutdown. You execute a SRVCTL command on any node and the results are returned to you.</p> <p>The GSD is installed on the nodes and is implemented on UNIX, Windows NT and Windows 2000 operating systems.</p>

The Global Services Daemon (GSD) enables EM or the SRVCTL utility to perform system management tasks. It is important that you do not kill this process.

The Oracle Intelligent Agent invokes SRVCTL to execute jobs. The multiple GSDs then receive a request from an SRVCTL command, and the results are returned to SRVCTL, then to the Agent, and then to the console as shown in [Figure 7-1](#):

Figure 7-1 SRVCTL Architecture for UNIX and Windows NT and Windows 2000



See Also: ■ *Oracle9i Real Application Clusters Administration* for information on using SRVCTL and *Oracle Intelligent Agent User's Guide* for information about the Oracle Intelligent Agent

Server Management Requirements

Each node must meet the requirements as shown in [Table 7-2](#) to perform Real Application Clusters-specific tasks by way of SRVM from the **Console**. These requirements should have been satisfied if you followed the database creation procedures described in Part I of this book, "[Setup and Configuration of Oracle9i Real Application Clusters](#)".

Table 7-2 Node Requirements

Node Requirements	References
Ensure Oracle9i Enterprise Edition , with Real Application Clusters, and the Oracle Intelligent Agent are installed on each node in the cluster.	Part I of this book, " Setup and Configuration of Oracle9i Real Application Clusters ".
Each instance's Oracle System Identifiers (sid) in the cluster database must be unique.	Part I of this book, " Setup and Configuration of Oracle9i Real Application Clusters ".
Each node must have a <code>tnsnames.ora</code> file configured with a net service name entry for: <ul style="list-style-type: none"> ■ The database ■ Each instance 	Part III of this book, " Oracle Enterprise Manager Installation Highlights for Real Application Clusters ".
The listener.ora file must define a listener for each node with: <ul style="list-style-type: none"> ■ A TCP/IP address for the listener ■ An entry for the <code>sid</code> of the node's instances in the <code>sid_LIST_listener_name</code> section. 	Part III of this book, " Oracle Enterprise Manager Installation Highlights for Real Application Clusters ".
The <code>oratab</code> file must be configured on UNIX for the agent to discover the database.	Part I of this book, " Setup and Configuration of Oracle9i Real Application Clusters ".
Make sure that the Global Services Daemon is running on each managed node using the <code>gsdctl stat</code> command by executing it from the <code>\$ORACLE_HOME/bin</code> directory. On Windows NT and Windows 2000 platforms, this daemon is called OracleGSDService and you should run the <code>gsdctl stat</code> from the <code>%ORACLE_HOME%\bin</code> directory.	Part I of this book, " Setup and Configuration of Oracle9i Real Application Clusters ".
The <code>srvcfg</code> cluster configuration file must have entries for the cluster database and instances.	Part I of this book, " Setup and Configuration of Oracle9i Real Application Clusters ".

Understanding the Oracle Enterprise Manager Setup

You can run the individual EM components on separate machines or combine components on separate machines to collaboratively manage the complete Oracle environment. You can also run the console in standalone mode as described under the following headings:

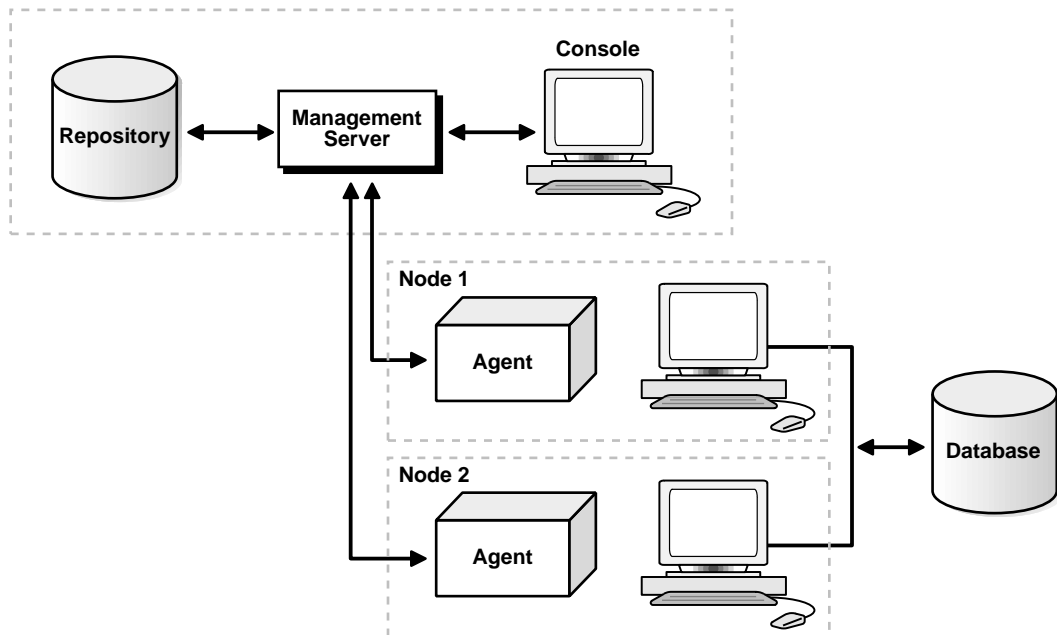
- [Using the Console, Management Server, and Repository on the Same Machine](#)
- [Using the Console, Management Server, and Repository on Separate Machines](#)
- [Notes for Running the Console in Standalone Mode](#)

Using the Console, Management Server, and Repository on the Same Machine

The Console, with the aid of the [Management Server](#), remotely manages the databases for both nodes as shown in [Figure 7-2](#). The Console and the Management Server are running on a Solaris, Windows NT, or Windows 2000 machine with an Oracle database installed that is only used as a repository. The Oracle Intelligent Agent does *not* need to be running unless you are managing the repository database using the EM job and event systems.

The nodes share an Oracle database. The repository is *not* created in this database, and Oracle Intelligent Agent runs on *both* nodes.

Figure 7-2 Console, Management Server, and Repository on the Same Machine

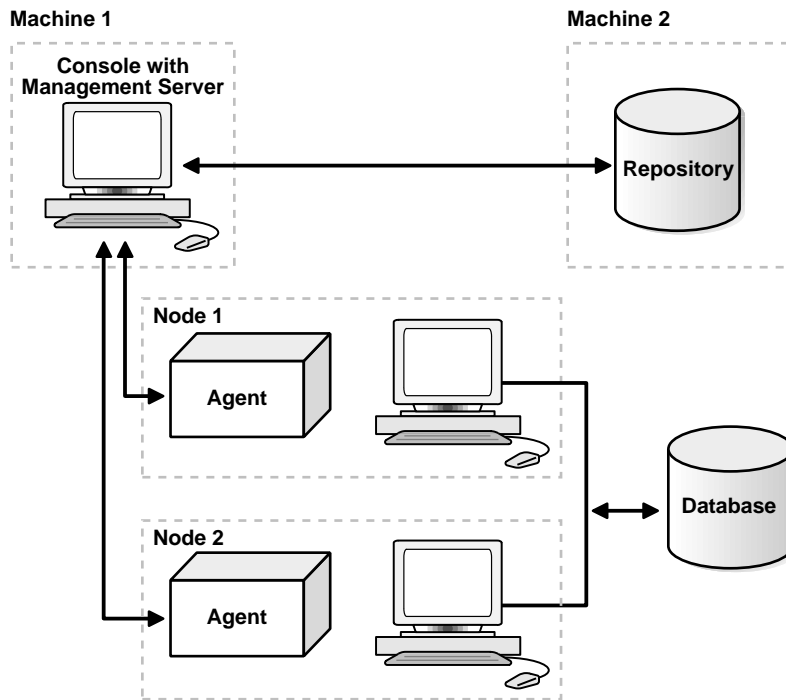


Using the Console, Management Server, and Repository on Separate Machines

The Repository is on a separate Solaris, Windows NT, or Windows 2000 machine with an Oracle database that is used solely as a repository as shown in [Figure 7-3](#). The Oracle Intelligent Agent does *not* need to be running on the machine that only hosts the repository.

The Management Server remotely manages the databases for the Console on a Windows machine. The Console is running on a Windows NT, Windows 95, Windows 98, or Windows 2000 machine.

The nodes run an Oracle database. The repository is *not* created in this database, and the Oracle Intelligent Agent runs on *both* nodes.

Figure 7-3 Console, Management Server, and Repository on Separate Machines

Notes for Running the Console in Standalone Mode

To run the EM console in standalone mode without a connection to the Management Server, manually enter the connection details for the cluster database instances that you want to manage. If you have a `tnsnames.ora` file, then you can use the cluster databases listed in it as targets.

If you have entered the connection details for the cluster database, the database is displayed as a single instance database target. In this case, the startup and shutdown menu items would only start and stop the instance to which the connection has been established.

Configuring Enterprise Manager for Real Application Clusters Management

This section describes the installation and configuration process for EM in Real Application Clusters.

Task 1: Oracle Enterprise Manager Installation

1. Install EM. Note that you can either install a complete Oracle9i database or Oracle9i Management and Integration. (EM is included in both software packages.)
2. Use the **Enterprise Manager Configuration Assistant (EMCA)** to create and load your version 2 repository. The repository is a set of tables in an Oracle database that stores data required by EM. If you install an Oracle9i Management and Integration, then EMCA runs automatically.
3. Start the Oracle Intelligent Agent on each of the nodes.
4. Start the Management Server.
5. Start the Enterprise Manager Console.

See Also:

- *Oracle Enterprise Manager Configuration Guide* for complete EM configuration instructions.
- *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, and Oracle9i Database Installation Guide for Windows* to install EM without Oracle Diagnostics Pack
- *Oracle Enterprise Manager Installation* to install EM and the Oracle Diagnostics Pack

Task 2: Specify Preferred Credentials for Nodes and Database

You must configure the EM with two types of credentials:

- Cluster Database credentials
- Node credentials

EM uses cluster database credentials when establishing connections to Real Application Clusters, such as when the cluster or database instance is expanded in

the Console's navigator. Startup and shutdown operations use the node credentials in addition to the cluster database credentials.

The cluster database credentials you configure must identify a valid DBA user with **SYSDBA** or **SYSOPER** privileges for the database on the target node. This enables you to expand the Database folder in the Navigator window and to perform connection and job execution operations. If you do not identify a valid database user, EM prompts you for this information each time you attempt to connect to the database.

You must also identify an operating system user to run jobs on particular nodes, such as starting or stopping an instance. Although you submit a job from the Console, the job scripts themselves reside on the Oracle Intelligent Agent on the nodes. For this reason, you must configure a user that has operating system access to the node.

To complete the installation process, you must complete the following tasks related to creating accounts, granting privileges, and setting credentials.

Task 3: Create an Operating System Account

On UNIX, this user can be the *oracle* account set up during the installation process.

On Windows NT or Windows 2000, create a Windows user account. You must be a member of the "Administrators" group and you must also select the **Logon on as a batch job** User Right policy.

See Also: Microsoft Corporation documentation at <http://www.microsoft.com> for details on setting up User Accounts, Group Membership, and User Right Policy information.

Task 4: Grant SYSDBA or SYSOPER Privileges to a Database User

Identify a current user, such as *SYSTEM*, or create a new user that will connect, start, and stop the database.

Once you identify a user, ensure it has *SYSDBA* or *SYSOPER* privileges. *SYSDBA* and *SYSOPER* privileges contain all the system privileges you need to manage the database.

To grant *SYSDBA* or *SYSOPER* privileges to a user, use the *GRANT* command:

```
SQL>GRANT sysdba to username;  
SQL>GRANT sysoper to username;
```

Task 5: Set Preferred Credentials in the Console

You must set user credentials for the cluster database, each cluster database instance, and each node.

To set credentials:

1. Choose Configuration > Preferences.
The Edit Administrator Preferences dialog box appears.
2. Click the **Preferred Credentials** tab to enter credentials for various target types.
3. Click a node (identified by the Node target type) and enter the operating system user name and password.
4. Perform Step 3 for each node in your cluster.
5. Click the cluster database identified by the target type

Enter a DBA user name and password that has `SYSDBA` or `SYSOPER` privileges for the target database, such as `SYSTEM/password`, and select **SYSDBA** or **SYSOPER** from the Role list.

Important: The `SYSDBA` or `SYSOPER` privilege is required for the cluster database, instance startup and shutdown, initialization parameter management, and resource plan management.

See Also:

- *Oracle Enterprise Manager Administrator's Guide* for general Console administration information
- *Oracle Enterprise Manager Configuration Guide* for optional Console configuration

Using Oracle Performance Manager for Real Application Clusters Management

Note: Oracle Performance Manager can be run with or without running EM. If you run this product as a standalone product, then you do not have to configure EM.

To configure Oracle Performance Manager, perform the following tasks:

Task 1: Start Oracle Performance Manager

Start Oracle Performance Manager in Standalone mode or from the Console.

See Also:

- *Oracle Enterprise Manager Administrator's Guide* for general Console administration information
- *Oracle Enterprise Manager Configuration Guide* for optional Console configuration

Task 2: Accessing Statistical Charts

You can access statistical charts from the Cluster Databases folder or from the Databases folder.

Cluster Database Instances Folder

If the login occurred from the Console or if you selected the **Login to the Oracle Management Server** option in the Performance Manager Login dialog box, then you can expand the Cluster Database Instances folder to display the instances from which the list of available charts can be obtained.

Note: When you are accessing a cluster database of version 8.1.7 and earlier, you will see charts with Oracle Parallel Server-specific names. When accessing a cluster database of version 9.0.1 and later, you will see charts referred to with Real Application Clusters-specific names.

Once you expand the Instances folder, expand Real Application Clusters.

If you did not specify credentials for the instances, as described in "[Task 2: Specify Preferred Credentials for Nodes and Database](#)" on page 7-8, then the Instances Logon dialog appears when you attempt to expand an instance.

See Also: *Oracle9i Real Application Clusters Deployment and Performance* for detailed information on the Performance Manager charts

Enter a database user and password.

Databases Folder

If the login occurred in standalone mode or if you selected the **Standalone, no repository connection** option in the Performance Manager Login dialog box, then you can access the Real Application Clusters charts from the Databases folder:

1. Click the **Databases** folder.
2. Choose File > Add New Service.

The Database Logon dialog prompts you for the required Oracle database credential.

3. Enter the required information, including a database user name and password, and a net service name in the Service field.
4. Expand Databases > Cluster Databases (or Parallel Servers, depending on the software version you are running).

See Also:

- *Oracle9i Real Application Clusters Deployment and Performance* for more information about viewing Real Application Clusters performance statistics
- *Getting Started with the Oracle Standard Management Pack* for general information about the Oracle Performance Manager application

Part III

The Real Application Clusters Installed Configuration and High Availability

Part III describes the configuration for Real Application Clusters and explains how to configure Oracle high availability features. The chapters in Part III are:

- [Chapter 8, "Understanding the Installed Configuration for Real Application Clusters"](#)
- [Chapter 9, "Configuring High Availability Features for Real Application Clusters"](#)

Understanding the Installed Configuration for Real Application Clusters

This chapter describes the installed configuration for **Real Application Clusters**. The topics in this chapter include:

- Understanding the Configured Environment
- Shared Configuration File
- UNIX Operating System Configurations
- Database Components Created Using the Database Configuration Assistant
- Managing Undo Tablespaces
- Initialization Parameter Files
- Configuring Service Registration-Related Parameters in Real Application Clusters
- Configuring the Listener File (listener.ora)
- Directory Server Access (ldap.ora File)
- Net Service Names (tnsnames.ora File)
- Profile (sqlnet.ora File)

Understanding the Configured Environment

The **Oracle Net Configuration Assistant** and the **Database Configuration Assistant (DBCA)** configure your environment to meet the requirements for database creation and **Oracle Enterprise Manager** discovery of Real Application Cluster databases.

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

Shared Configuration File

The Database Configuration Assistant uses a shared configuration file for storing the configuration for the cluster databases that it creates. This file is a shared file in a cluster file system environment. If you do not use a cluster file system, then you must make this file a shared **raw device** in UNIX environments, or a shared logical partition in Windows environments.

The **Oracle Universal Installer (OUI)** automatically initializes this shared configuration file by executing the `srvconfig -init` command. If this initialization does not complete, manually initialize the shared configuration file by executing the `srvconfig -init` command.

You can also use the `srvconfig` command to import or export the contents of the shared configuration file to or from a text file. In UNIX environments you can also use `srvconfig` to convert a pre-Oracle9i Oracle Parallel Server `dbname.conf` file to the shared configuration file. Refer to one of the following sections for platform-specific information on the `srvconfig` command:

- [Executing srvconfig on UNIX Platforms](#)
- [Executing srvconfig on Windows Platforms](#)

Executing srvconfig on UNIX Platforms

Executing the `srvconfig -init` command syntax requires that the `srvConfig.loc` file exists and that it contains the following entry:

```
srvconfig_loc=path_name
```

Where `path_name` is the complete path name for the shared configuration file. The `srvConfig.loc` file usually resides in the `/var/opt/oracle` directory. However, on HPUX, it resides in the `/etc` directory.

Executing `srvconfig` on Windows Platforms

When you execute the `srvconfig -init` command on Windows NT and Windows 2000 you must have a symbolic link named `svcfg` that you created using the **Object Link Manager (OLM)**. This symbolic link must point to the correct disk partition.

UNIX Operating System Configurations

This section describes the `oratab` file configuration.

The `oratab` File on UNIX

Oracle creates an entry for the Real Application Clusters database in the `oratab` file. Oracle Enterprise Manager uses this file during **service discovery** to determine the name of the Real Application Clusters database as well whether it should be auto-started on restart. The Real Application Clusters database entry has the following syntax:

```
db_name:$ORACLE_HOME:N
```

Where *db_name* is the database name for your Real Application Clusters database, `$ORACLE_HOME` is the directory path to the database, and `N` indicates that the database should not be started at restart time. A sample entry for a database named `db` is:

```
db:/private/system/db:N
```

Database Components Created Using the Database Configuration Assistant

This section describes the database components created by the Database Configuration Assistant (DBCA). The topics in this section include:

- [Tablespaces and Datafiles](#)
- [Redo Log Files](#)
- [Control Files](#)

Tablespaces and Datafiles

An Oracle database for both single-instance and cluster database environments is divided into smaller logical areas of space known as tablespaces. Each tablespace corresponds to one or more datafiles stored on a disk. [Table 8–1](#) shows tablespace names used by a Real Application Clusters database and the types of data they contain:

Table 8–1 *Tablespace Names used by Oracle Real Application Clusters Databases*

Tablespace Name	Contents
SYSTEM	Consists of the data dictionary, including definitions of tables, views, and stored procedures needed by the database. Information in this area is maintained automatically. The SYSTEM tablespace is present in all Oracle9i databases.
USER	Consists of application data. As you create and enter data into tables, Oracle fills this space with your data.
TEMP	Contains temporary tables and indexes created during SQL statement processing. You may need to expand this tablespace if you are executing a SQL statement that involves significant sorting, such as ANALYZE COMPUTE STATISTICS on a very large table, or the constructs GROUP BY, ORDER BY, or DISTINCT. Note: A future release of Oracle will limit the use of the ANALYZE statement to collect optimizer statistics.
UNDOTBS_n	These are the undo tablespaces for each instance that the Database Configuration Assistant creates for automatic undo management .
RBS	If you do not use automatic undo management, then Oracle uses the RBS tablespace for the rollback segments.
INDX	Stores indexes associated with the data in the USER tablespace.
TOOLS	Stores tables for Oracle Enterprise Manager.
DRSYS	Consists of data for Oracle9i <i>interMedia</i> Text.

Table 8–1 (Cont.) Tablespace Names used by Oracle Real Application Clusters

Tablespace Name	Contents
EXAMPLE	Stores the Oracle9i Sample Schemas
CWMLITE	Stores the OLAP files.
XML	To accommodate XML features.
ODM	To accommodate Oracle Data Mining features.

You cannot alter the tablespace names when using the preconfigured database configuration options from the Oracle Universal Installer. However, you can change the names of the tablespaces when using the **Customized** database creation method.

As mentioned, each tablespace has one or more datafiles. The datafile names created by the preconfigured database configuration options vary by operating system. UNIX prompts you to set the file names. Windows NT and Windows 2000 use the **symbolic link names** shown in [Table 8–2](#):

Table 8–2 Windows NT and Windows 2000 Symbolic Link Names

Windows NT and Windows 2000 Symbolic Link Names	Tablespaces
<code>db_name_system</code>	SYSTEM
<code>db_name_spfile</code>	SPFILE
<code>db_name_users</code>	USERS
<code>db_name_temp</code>	TEMP
<code>db_name_undotbs1</code>	UNDOTBS1
<code>db_name_undotbs2</code>	UNDOTBS2
<code>db_name_rbs</code>	RBS (optional)
<code>db_name_example</code>	EXAMPLE
<code>db_name_cwmlite</code>	CWMLITE
<code>db_name_xml</code>	XML
<code>db_name_odm</code>	ODM
<code>db_name_indx</code>	INDX
<code>db_name_tools</code>	TOOLS

Table 8–2 (Cont.) Windows NT and Windows 2000 Symbolic Link Names

Windows NT and Windows 2000 Symbolic Link Names	Tablespaces
<i>db_name_drsys</i>	DRSYS
<i>db_name_control1</i>	Control File 1
<i>db_name_control2</i>	Control File 2
<i>srvcfg</i>	SRVM Configuration
<i>db_name_redo_thread_number</i>	Redo Log Files
Where <i>thread</i> is the thread ID of the instance and <i>number</i> is the log number (1 or 2) for the instance.	Each instance must have two redo log files. If the database name is <i>db</i> , the link names for the first instance should be: <i>db_redo1_1</i> <i>db_redo1_2</i> The link names for the second instance should be: <i>db_redo2_1</i> <i>db_redo2_2</i>

You can specify different symbolic names with the Customized database configuration option.

Redo Log Files

Each instance is configured with at least two redo log files that are stored in the shared files. If you have a cluster file system, then these files are file system files. If you do not have a cluster file system, then these files are raw devices. The redo log files' names created with the preconfigured database configuration options vary by operating system.

You must enter the raw device names on UNIX unless you are using a cluster file system. On Windows NT and Windows 2000 use symbolic link names of `\\.\db_name_thread_number`, where *thread* is the thread ID of the instance, and *number* is the number, 1 or 2, of the redo log file.

To use the Customized database creation method, locate the redo log files in the Database Storage page and replace their default filenames to be the correct raw device names or symbolic link names.

Note: Where the notation *db_name* appears in the previous example and throughout this chapter, it refers to the database name you entered when prompted by the DBCA, or to the entry you made for the `DATABASE` keyword of the `CREATE DATABASE` statement.

Control Files

The database is configured with two control files that are stored on the raw devices. The control files' names created by the preconfigured database configuration options vary by operating system. UNIX prompts you to set the file names. Windows NT and Windows 2000 use symbolic link names of *db_name_control1* and *db_name_control2*. The Customized database creation method prompts you to specify control file names or symbolic link names.

Managing Undo Tablespaces

Oracle stores rollback or *undo* information in undo tablespaces. To manage undo tablespaces, Oracle Corporation recommends that you use **automatic undo management**. Automatic undo management is an automated undo tablespace management mode that is easier to administer than manual undo management.

If you are not using automatic undo management, then the undo rollback segments created for the Customized database creation type have names in the format of *rbstthread_number*, where *thread* is the thread ID of the instance, and *number* is the number, 1 or 2, of the rollback segment.

See Also: *Oracle9i Real Application Clusters Administration* for more information on managing undo tablespaces

Initialization Parameter Files

Oracle Corporation recommends using the **server parameter file**. This file resides on the server on the shared disk; all instances in a **cluster database** can access this parameter file. This file is a binary file that you cannot directly modify.

To change the values of parameters in this file, use Oracle Enterprise Manager or the `ALTER SYSTEM SET` syntax. You can also use the traditional client-side parameter files.

See Also: [Chapter 6, "Configuring the Server Parameter File in Real Application Clusters Environments"](#) for more information on the creation and use of parameter files

Configuring Service Registration-Related Parameters in Real Application Clusters

Two of the key benefits of Real Application Clusters are connection load balancing and failover. Real Application Clusters extends the ability of single-instance Oracle database load balancing, where connections are distributed among local dispatchers, to the balancing of connections among *all instances* within a cluster database. In addition, Real Application Clusters provides failover by configuring multiple listeners on multiple instances to manage client connection requests for the same database service. Connection load balancing and failover increase availability by taking advantage of the redundant resources within a cluster database. These features, however, require cross-node registration.

Cross-node registration in Real Application Clusters occurs when an instance's PMON process not only registers with the local listener, but when it also registers with all other listeners. Thus, all instances in the cluster register with all listeners in the cluster. This enables all listeners to manage connections across all instances for both load balancing and failover.

Cross-node registration requires configuration of the `LOCAL_LISTENER` and `REMOTE_LISTENER` initialization parameters. The `LOCAL_LISTENER` parameter identifies the local listener and the `REMOTE_LISTENER` parameter identifies the global list of listeners of the instance. Both of these parameters are dynamic initialization parameters. Oracle commonly changes the setting for `REMOTE_LISTENER` dynamically when you reconfigure your cluster, for example, when you add or delete nodes.

By default, the DBCA only configures dedicated servers. However, if you select the Shared server option on the DBCA, then Oracle configures the shared server. In this case, Oracle uses both dedicated and shared server processing. When shared servers are configured, the `DISPATCHER` parameter is specified as in the following example:

```
DISPATCHERS="(protocol=tcp)"
```

If the `DISPATCHERS` initialization parameter does not specify the `LISTENER` attribute as in the previous example, then the PMON process registers information for all dispatchers with the listeners specified by the `LOCAL_LISTENER` and `REMOTE_LISTENER` parameters.

However, when the `LISTENER` attribute is specified, then the PMON registers dispatcher information with the listeners specified by the `LISTENER` attribute. In this case, setting the `LISTENER` attribute overrides the settings for `LOCAL_LISTENER` and `REMOTE_LISTENER` for the specified dispatchers as in the following example:

```
DISPATCHERS="(protocol=tcp)(listener=listeners_db_name)"
```

See Also: *Oracle9i Net Services Administrator's Guide* for further information about cross-node registration, shared and dedicated server configurations, and connection load balancing

Configuring the Listener File (listener.ora)

You can configure two types of listeners in the [listener.ora](#) file as described under the following headings:

- [Nondefault Listeners](#)
- [Multiple Listeners](#)
- [How Oracle Uses the Listener \(listener.ora File\)](#)

Nondefault Listeners

If you configured a listener that does not use the default listener address of TCP/IP port 1521, then the Database Configuration Assistant automatically configures the `LOCAL_LISTENER` parameter in the [initialization parameter file](#) as follows, where `listener_sid` is resolved to a listener address through either a `tnsnames.ora` file on the machine, or through the Oracle Names Server:

```
sid.local_listener=listener_sid
```

Multiple Listeners

If the DBCA detects more than one listener on each node, it displays a list of the listeners. You can select one of these listeners. If you select a nondefault listener, then the `LOCAL_LISTENER` parameter is set in the initialization parameter file, as described previously in "[Nondefault Listeners](#)".

How Oracle Uses the Listener (listener.ora File)

Services coordinate their sessions with the help of a listener, a process on the server that receives connection requests on behalf of a client application. Listeners are configured to “listen on” protocol addresses for a database service or non-database service.

Protocol addresses are configured in the listener configuration file, `listener.ora`, for a database service or a non-database service. Clients, configured with the same addresses, can connect to a service through the listener.

During a preconfigured database configuration install, the Oracle Net Configuration Assistant creates and starts a default listener called `LISTENER`. The listener is configured with default protocol listening addresses for the database and **external procedures**. During a **Customized** installation, you are prompted to create at least one listener with the Oracle Net Configuration Assistant. The listener is configured to listen on one protocol address you specify, as well as an address for **external procedures**.

Note: If your platform supports a cluster file system, the default name for the listener is `listener_node_name`,

Both installation modes configure service information about the Real Application Clusters database and external procedures. An Oracle9i Release 2 (9.2) database service automatically registers its information with the listener, such as its **service name**, instance name(s), and load information.

This feature, called **service registration**, does not require configuration in the `listener.ora` file. However, Oracle Enterprise Manager tools require static service configuration in the `listener.ora` file to discover the database instance. The database service information includes the `ORACLE_HOME` of the database instance and the Oracle System Identifier (**sid**) information of the instance.

After listener creation, the listener is started by Oracle Net Configuration Assistant. A sample `listener.ora` file with an entry for an instance named `db1` is:

```
listener=
  (description=
    (address=(protocol=ipc)(key=extproc))
    (address=(protocol=tcp)(host=db1-server1)(port=1521)))
sid_list_listener=
  (sid_list=
    (sid_desc=
      (sid_name=plsextproc)
```



```
(oracle_home=/private/system/db)
(program=extproc)
(sid_desc=
  (oracle_home=/private/system/db)
  (sid_name=db1)))
```

Notice that the second `sid_DESC` entry for the instance does not use the `GLOBAL_DBNAME` parameter entry; this prevents the disabling of **transparent application failover (TAF)**. This entry is typical for a `listener.ora` file entry for a single-instance database, as shown in the following:

```
(sid_desc=
  (global_dbname=sales.us.acme.com)
  (sid_name=sales)
  (oracle_home=/private/system/db)))
```

Note: In Real Application Clusters environments, the `GLOBAL_DBNAME` parameter disables **connect-time failover** or **transparent application failover (TAF)**. Oracle Corporation strongly recommends against adding this parameter to your `listener.ora` file.

Listener Registration and PMON Discovery

When a listener starts *after* the Oracle instance starts, and the listener is listed for service registration, registration does not occur until the next time the PMON discovery routine executes. By default, this is 60 seconds later.

This problem occurs when a listener is started after the Oracle instance and every time that listener fails and is restarted. To override the 60 second delay, you can use the system-level SQL statement `ALTER SYSTEM REGISTER`. This statement forces PMON to register the service immediately.

Oracle Corporation recommends that you create a script to execute this statement immediately after starting the listener. If you execute this statement while the listener is up and the instance is already registered, or while the listener is down, then the statement has no effect.

See Also: *Oracle9i Net Services Administrator's Guide* for further information about the listener and the `listener.ora` file

Directory Server Access (ldap.ora File)

If you configure access to an **Lightweight Directory Access Protocol (LDAP)**-compliant directory server with the Oracle Net Configuration Assistant during a Customized installation, an `ldap.ora` file is created. The `ldap.ora` file contains the following types of information:

- Type of directory
- Location of the directory
- **administrative context** from which this server can look up, create, and modify a **net service name** and the database service entries

See Also: *Oracle9i Net Services Administrator's Guide* for further information about directory naming configuration and directory server access configuration

Net Service Names (tnsnames.ora File)

A `tnsnames.ora` file is created on each node and an LDAP directory (if configured during an Customized installation) is configured with **net service names**. A **connect identifier** is an identifier that maps to a **connect descriptor**. A connect descriptor contains the following information:

- Network route to the service, including the location of the listener through a protocol address
- `SERVICE_NAME` for an Oracle release 8.1 or later, or `sid` for pre-8.1 Oracle releases

Note: The `SERVICE_NAME` parameter you use in `tnsnames.ora` is singular because you can only specify one service name.

The Database Configuration Assistant creates net service names for the connections as shown in [Table 8-3](#):

Table 8-3 Connections for Net Service Names

Net Service Name Type	Description
Database connections	<p>Clients that connect to any instance of the database use the Net Service Name entry for the database. This entry also enables Oracle Enterprise Manager to discover a Real Application Clusters database.</p> <p>A listener address is configured for each node that runs an instance of the database. The <code>LOAD_BALANCE</code> option causes Oracle to choose the address randomly. If the chosen address fails, then the <code>FAILOVER</code> option causes the connection request to fail over to the next address. Thus, if an instance fails, then clients can still connect using another instance.</p> <p>In the following example, <code>db.us.oracle.com</code> is used by the client to connect to the target database, <code>db.us.oracle.com</code>.</p> <pre>db.us.acme.com= (description= (load_balance=on) (address=(protocol=tcp)(host=db1-server)(port=1521) (address=(protocol=tcp)(host=db2-server)(port=1521) (connect_data= (service_name=db.us.acme.com)))</pre> <p>Note: <code>FAILOVER=ON</code> is set by default for a list of addresses. Thus, you do not need to explicitly specify the <code>FAILOVER=ON</code> parameter.</p>
Instance connections	<p>Clients that connect to any instance of the database use the Net Service Name entry for the database. This entry, for example, enables Oracle Enterprise Manager to discover the instances in the cluster. These entries are also used to start and stop instances.</p> <p>In the following example, <code>db1.us.acme.com</code>, is used by Oracle Enterprise Manager to connect to an instance named <code>db1</code> on <code>db1-server</code>:</p> <pre>db1.us.acme.com= (description= (address=(protocol=tcp)(host=db1-server)(port=1521)) (connect_data= (service_name=db.us.acme.com) (instance_name=db1)))</pre>

Table 8–3 (Cont.) Connections for Net Service Names

Net Service Name Type	Description
Remote listeners	<p>As discussed in "Configuring Service Registration-Related Parameters in Real Application Clusters" on page 8-8, the</p> <p>Whether using shared servers or dedicated servers, the list of remote listeners is supplied using the <code>REMOTE_LISTENERS</code> parameter, for example:</p> <pre>REMOTE_LISTENERS=listeners_<dbname></pre> <p>This enables the instance connect to remote listeners on the other nodes, and <code>listeners_db_name</code> is resolved through the <code>tnsnames.ora</code> file.</p> <p>In the following example, <code>listeners_db.us.acme.com</code> is resolved to list of listeners available in the cluster database as shown in the following example:</p> <pre>listeners_db.us.acme.com= (address=(protocol=tcp)(host=db1-server)(port= 1521)) (address=(protocol=tcp)(host=db2-server)(port=1521))</pre> <p>The instance uses this list to determine the addresses of the remote listeners with which to register its information.</p>
Nondefault listeners	<p>As discussed in "Nondefault Listeners" on page 8-9 and "Multiple Listeners" on page 8-9, the <code>LOCAL_LISTENER</code> parameter is set in the <code>init.ora</code> file if a nondefault listener is configured.</p> <pre>sid.local_listener=listener_sid</pre> <p><code>listener_sid</code> is resolved to a listener address.</p> <p>In the following sample, <code>listener_db1.us.acme.com</code> is resolved to the nondefault listener address:</p> <pre>listener_db1.us.acme.com= (address=(protocol=tcp)(host=db1-server)(port= 1421))</pre>
External procedures	<p>An entry for connections to external procedures. This enables an Oracle9i database to connect to external procedures.</p> <pre>extproc_connection_data.us.acme.com= (description= (address_list= (address=(protocol=ipc)(key=extproc0)) (connect_data= (sid=plsextproc)))</pre>

Example 8-1 Example tnsnames.ora File

The following is a sample `tnsnames.ora` file that is created during a preconfigured database configuration install:

```
db.us.acme.com=
  (description=
    (load_balance=on)
    (failover=on)
    (address_list=
      (address=(protocol=tcp)(host=db1-server)(port=1521))
      (address=(protocol=tcp)(host=db2-server)(port=1521)))
    (connect_data=
      (service_name=db.us.acme.com)))

db1.us.acme.com=
  (description=
    (address=(protocol=tcp)(host=db1-server)(port=1521))
    (connect_data=
      (service_name=db.us.acme.com)
      (instance_name=db1)))

db2.us.acme.com=
  (description=
    (address=(protocol=tcp)(host=db2-server)(port=1521))
    (connect_data=
      (service_name=db.us.acme.com)
      (instance_name=db2)))

listeners_db.us.acme.com=
  (address=(protocol=tcp)(host=db1-server)(port=1521))
  (address=(protocol=tcp)(host=db2-server)(port=1521))

extproc_connection_data.us.acme.com=
  (description=
    (address_list=
      (address=(protocol=ipc)(key=extproc)))
    (connect_data=
      (sid=plsextproc)
      (presentation=RO)))
```

See Also: *Oracle9i Net Services Administrator's Guide* for further information about the `tnsnames.ora` file

Profile (sqlnet.ora File)

The **sqlnet.ora** file is automatically configured with:

- Computer's domain.

This domain is automatically appended to any unqualified net service name or service name. For example, if the default domain is set to `us.acme.com`, Oracle resolves `db` in the connect string `CONNECT scott/tiger@db as:`
`db.us.acme.com`.

- A **naming method** the server can use to resolve a name to a connect descriptor.

The order of naming methods is as follows: directory naming (for the Customized installations only), `tnsnames.ora` file, Oracle Names server, and host naming.

The following is a sample `SQLNET.ORA` file created during a preconfigured database configuration install:

```
names.default_domain=us.acme.com
names.directory_path=(tnsnames, onames,hostname)
```

See Also: The *Oracle9i Net Services Administrator's Guide* for further information about the `sqlnet.ora` file

Configuring High Availability Features for Real Application Clusters

This chapter describes how to configure the high availability features of **Real Application Clusters**. The topics in this chapter are:

- [Transparent Application Failover](#)
- [Primary/Secondary Instances](#)
- [Real Application Clusters Guard II](#)
- [Configuring Clients for Real Application Clusters](#)

Transparent Application Failover

This section discusses the Real Application Clusters-specific aspects of **transparent application failover (TAF)**. This section covers the following topics:

- [FAILOVER_MODE Parameters](#)
- [TAF Implementation](#)
- [Transparent Application Failover Verification](#)

To enable TAF, manually configure a **net service name** in the `CONNECT_DATA` portion of the **connect descriptor**. In the net service name you can include the `FAILOVER_MODE` and `INSTANCE_ROLE` parameters.

TAF instructs **Oracle Net** to move a connection that has failed to a different **instance**. This enables the user to continue working by using the new connection as if the original connection had never failed.

FAILOVER_MODE Parameters

To configure TAF, include `FAILOVER_MODE` parameter settings in the `CONNECT_DATA` portion of a connect descriptor. There are several sub-parameters you can use to specify `FAILOVER_MODE`. For example, you can specify the type of failover and speed at which Oracle should process it, as well as the retry behavior Oracle should use.

See Also: *Oracle9i Net Services Administrator's Guide* for detailed information about the sub-parameters you can use for `FAILOVER_MODE` and the *Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration* for information on configuring TAF for Oracle Real Application Clusters Guard

TAF Implementation

Depending on the `FAILOVER_MODE` parameters, you can implement TAF using any of the following methods:

- [Implementing TAF with Connect-Time Failover and Client Load Balancing](#)
- [Retrying Connections](#)
- [Pre-Establishing Connections](#)

Implementing TAF with Connect-Time Failover and Client Load Balancing

You can implement TAF with connect-time failover and **client load balancing** for multiple addresses. In the following example, Oracle Net connects randomly to one of the **listener** addresses on **node** db1 or db2. If the instance fails after the connection, Oracle Net fails over to the other node's instance, preserving any SELECT statements in progress.

```
db.us.acme.com=
(description=
  (load_balance=on)
    /* only connect time load balancing and connection load balancing */
  (failover=on)          /* only connect time failover */
(address=
  (protocol=tcp)
  (host=db1-server)
  (port=1521))
(address=
  (protocol=tcp)
  (host=db2-server)
  (port=1521))
(connect_data=
  (service_name=db.us.acme.com)
  (failover_mode=
    (type=select)
    (method=basic))))
```

See Also: *Oracle9i Net Services Administrator's Guide* for more information about and examples of load balancing

Retrying Connections TAF also provides the ability to automatically retry connecting with the `RETRIES` and `DELAY` parameters if the first connection attempt fails. In the following example, Oracle Net attempts to connect to the listener on `db1-server`. If the initial connection fails, it fails over to addresses in the description lists.

```
db.us.acme.com=
(description=
(address=
(protocol=tcp)
(host=db1-server)
(port=1521))
(address=
(protocol=tcp)
(host=db2-server)
(port=1521))
(connect_data=
(service_name=db.us.acme.com)
(failover_mode=
(type=select)
(method=basic)
(retries=20)
(delay=15)))
```

See Also: *Oracle9i Net Services Administrator's Guide* for detailed information about `FAILOVER_MODE` sub-parameters

Pre-Establishing Connections A backup connection can be pre-established. The initial and backup connections must be explicitly specified. In the following example, Oracle Net connects to the listener on `db1-server`. If `db1-server` fails after the connection, Oracle Net fails over to `db2-server`, preserving any `SELECT` statements in progress. If the pre-connect to the failed instance does not succeed at connect time, then fail back to this instance is no longer possible.

```
db.acme.com=
(description=
(address=
(protocol=tcp)
(host=db1-server)
(port=1521))
(connect_data=
(service_name=db.us.acme.com)
(instance_name=db1-server)
(failover_mode=
(backup=db2.acme.com)
```

```

        (type=select)
        (method=preconnect)))
db2.acme.com=
(description=
(address=
(protocol=tcp)
(host=db2-server)
(port=1521))
(connect_data=
(service_name=db.us.acme.com)
(instance_name=db2-server)
(failover_mode=
(backup=db1.acme.com)
(type=select)
(method=preconnect))
))

```

Transparent Application Failover Verification

Use the `V$SESSION` view to obtain information about the connected clients and their TAF status. For example, query the `FAILOVER_TYPE`, `FAILOVER_METHOD`, and `FAILED_OVER` columns to verify that you have correctly configured TAF as in the following SQL statement:

```

SELECT MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER, COUNT(*)
FROM V$SESSION
GROUP BY MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER;

```

The output before failover resembles the following:

MACHINE	FAILOVER_TYPE	FAILOVER_M	FAI	COUNT(*)
db1	NONE	NONE	NO	11
db2	SELECT	PRECONNECT	NO	1

The output after failover is:

MACHINE	FAILOVER_TYPE	FAILOVER_M	FAI	COUNT(*)
db2	NONE	NONE	NO	10
db2	SELECT	PRECONNECT	YES	1

Note: You can monitor each step of TAF using an appropriately configured `OCI_TAF_CALLBACK` function.

See Also:

- *Oracle Call Interface Programmer's Guide*
- *Oracle9i Database Reference* for more information about the `V$SESSION` view

Primary/Secondary Instances

The Primary/Secondary Instances feature specifies that the primary instance accepts *primary* connections and the secondary instance only accepts *secondary* connections. You can only implement this feature for two-instance Real Application Clusters environments. This section describes the primary and secondary instance feature in more detail and contains the following topics:

- [Overview of Primary and Secondary Instances](#)
- [Initialization Parameter Configuration](#)
- [Client Configuration](#)
- [Listener Configuration](#)
- [Connecting to Secondary Instances](#)
- [Warming the Library Cache on the Secondary Instance](#)

Overview of Primary and Secondary Instances

An instance is the primary instance when `ACTIVE_INSTANCE_COUNT=1` is set in the initialization file and it has been started first. The primary instance registers its status and database service information with its local listener through dynamic **service registration**.

If you configure **shared server** with the `LISTENER` attribute, the primary instance can also register with the secondary instance's listener. The `LISTENER` parameter can specify a listener name alias for the listener to which the **dispatcher(s)** register information. This is resolved to a list of listener addresses through a **naming method**, such as a `tnsnames.ora` file.

This enables the primary instance to accept connections from its local listener, as well as from the secondary instance's listener. A secondary instance registers with its local listener as a secondary instance, and the secondary instance has its `ACTIVE_INSTANCE_COUNT` set to 1 in the `init.ora` file.

If the primary instance fails, the secondary instance assumes the primary role and registers with its listeners. When the failed instance re-starts, it starts up as the secondary instance.

Clients connected to the failed primary instance are failed over to the secondary instance if you have configured TAF. Clients connecting to the Real Application Clusters database after the primary instance fails are automatically routed to the secondary instance.

See Also:

- ["Transparent Application Failover"](#) on page 9-2 to configure TAF
- *Oracle9i Real Application Clusters Concepts* for conceptual information about primary and secondary instances

Initialization Parameter Configuration

To enable primary and secondary instance configuration, configure the initialization parameter file by setting the `ACTIVE_INSTANCE_COUNT` parameter to 1 on both instances.

Client Configuration

Oracle Corporation recommends configuring a connect descriptor on clients that use an address list containing the listener addresses for the primary instance and the secondary instance. Set the `LOAD_BALANCE` parameter to `OFF` because all client connections can only go to the primary instance. `FAILOVER` is set to `ON` by default for a list of addresses, so it does not need to be explicitly specified. An example of the client configuration follows:

```
db.us.acme.com=
(description=
  (load_balance=off) /* connection load balancing */
  (address=(protocol=tcp)(host=db1-server)(port=1521))
  (address=(protocol=tcp)(host=db2-sevrer)(port=1521))
  (connect_data=
    (service_name=db.us.oracle.com)))
```

Oracle does not recommend setting `LOAD_BALANCE=ON`. If you do, half of the connections attempt to connect to the listener on the secondary instance that fails to provide connections. The client then tries the listener on the primary instance, which succeeds.

See Also:

- *Oracle9i Net Services Administrator's Guide* to configure a connect descriptor
- *Oracle9i Net Services Administrator's Guide* to configure an address list and multiple address options, including **connect-time failover** and client load balancing

Listener Configuration

Remove the static service information `sid_LIST_listener_name` entry from the [listener.ora](#) file. This way, the listener only uses information obtained from dynamic service registration. For example, the `sid_list_listener` entry has been removed from the `listener.ora` file as shown in [Table 9-1](#):

Table 9-1 *Listener.ora Example without sid_list_listener Entry*

Old listener.ora File	Modified listener.ora File
<pre>listener= (description= (address= (protocol=tcp) (host=db1-server) (port=1521))) sid_list_listener= (sid_desc= (oracle_home=/private/system/db) (sid_name=db1))</pre>	<pre>listener= (description= (address= (protocol=tcp) (host=db1-server) (port=1521)))</pre>

Connecting to Secondary Instances

In some situations, you may wish to connect to the secondary instance even when the primary instance is active. For example, you may want to perform a batch operation on the database. To do this, use the `INSTANCE_ROLE` parameter in the connect data portion of the connect descriptor to configure explicit secondary instance connections.

The optional `INSTANCE_ROLE` parameter in the `CONNECT_DATA` section of a connect descriptor enables you to specify connections to primary or secondary instances. This parameter is useful for explicitly connecting to primary or secondary instances and for using Transparent Application Failover (TAF) to pre-connect to secondary instances.

See Also: *Oracle9i Net Services Administrator's Guide* for more detailed information and examples of `INSTANCE_ROLE`

Warming the Library Cache on the Secondary Instance

Maintaining information about frequently executed SQL and PL/SQL statements in the library cache improves performance. In Real Application Clusters primary and secondary instance configurations, the library cache associated with the primary instance contains up-to-date information. During failover, the benefit of that information is lost unless the library cache on the secondary instance was previously populated.

Use the `DBMS_LIBCACHE` package to transfer information in the library cache of the primary instance to the library cache of the secondary instance. This process is called **warming the library cache**. It improves performance immediately after failover because the new primary library cache does not need to be populated with parsed SQL statements and compiled PL/SQL units.

See Also: *Oracle9i Real Application Clusters Real Application Clusters Guard I - Concepts and Administration* for more information about this feature and *Oracle9i Supplied PL/SQL Packages and Types Reference* for more information about using `DBMS_LIBCACHE`

Real Application Clusters Guard II

The installation and setup of active/active cluster configurations is greatly simplified with Real Application Clusters Guard II. However, Real Application Clusters Guard II is *not* merely a two-node configuration. Rather, the size of a Real Application Clusters Guard II environment is only limited by the limitations of your hardware and operating system.

In Real Application Clusters Guard II, all instances are active and are able to support services. Instances are brought in automatically to support a service while they are also available to support other services.

A **service** in Real Application Clusters Guard II is a group of related business functions. The client selects a database service as part of the connection request

where the middle-tier specifies the `SERVICE_NAME` in the `tnsnames.ora` file's connect data description for each server in the connection pool. The service has associated functions—for statistics aggregation, for performance and resource consumption, and for configuring, measuring, and comparing service thresholds and workload distributions based on service levels and priority.

The notion of database service provides a single-system image for managing competing applications that run within a single instance and across multiple instances. The manageability features of the Real Application Clusters Guard II environment allow you to more easily control the instances on which applications run as well as their failover properties and failover targets.

See Also: *Oracle9i Real Application Clusters Guard II Concepts, Installation, and Administration* on the Real Application Clusters Guard II software CD

Configuring Clients for Real Application Clusters

This section describes client configuration issues not covered by the database creation process. You should configure the client with a net service name for the database. This entry should have an address list of all the listeners in the database. Additionally, set the connect-time failover and **client load balancing** options.

If the first listener fails, connect time failover instructs the client to failover to the next listener in the address list. Client load balancing instructs the client to randomly select a listener address. This randomization distributes the load to avoid overburdening a single listener.

There are two cases in which a client attempts to connect to another address:

- If the listener is down
- If the listener is up, but it has not received registration for the given `SERVICE_NAME`, so the instance is down

Note : The PMON process waits 60 seconds to reconnect to a failed listener. You can use the `ALTER SYSTEM REGISTER SQL` statement to force a reconnection attempt to occur immediately. You can only execute this statement, however, when the listener is running.

The second case implies that the client only attempts to connect to the next listener if the first listener fails. The client also tries the next listener if the first listener is up, but does not know about the `SERVICE_NAME` given in `CONNECT DATA`.

Together, connect-time failover and client load balancing instruct the client to randomly choose an address. If the chosen address fails, the connection request fails over to the next address. If an instance fails, the client can connect using another instance.

Warning: Do not set `GLOBAL_DB_NAME` in `listener.ora` because using this parameter disables Connect-time Failover and Transparent Application Failover.

Implementation of Client Configurations

To control how the client executes these connection attempts, configure multiple listening addresses and use `FAILOVER=ON` and `LOAD_BALANCE=ON` for the address list. For example:

```
db.us.acme.com=  
(description=  
  (load_balance=on)  
  (failover=on)  
  (address=(protocol=tcp)(host=db1-server)(port=1521))  
  (address=(protocol=tcp)(host=db2-server)(port=1521))  
  (connect_data=  
    (service_name=db.us.acme.com)))
```

See Also:

- *Oracle9i Net Services Administrator's Guide* to configure a connect descriptor
- *Oracle9i Net Services Administrator's Guide* to configure an address list and multiple address options, including connect-time failover and client load balancing

Note: Client load balancing may not be useful if you implement application partitioning.

Testing the Oracle Net Configuration

To ensure the files are configured correctly:

1. On any node or client machine, connect to an instance:

```
CONNECT SYSTEM/password@net_service_name
```

Oracle displays a "Connected" message.

If there is a connection error, troubleshoot your installation. Typically, this is a result of a problem with the IP address, host name, service name, or instance name.

2. On one node, increase Miller's salary by \$1000 and commit the change:

```
UPDATE EMP  
set sal = sal + 1000  
where ename = 'miller';  
commit;
```

3. On the other nodes, select the EMP table again:

```
SELECT * FROM EMP;
```

Miller's salary should now be \$2,300, indicating that all the instances can access the records in the database.

Part IV

Migrating to Real Application Clusters

Part IV provides information about migrating to Real Application Clusters. The chapter in Part IV is:

- [Chapter 10, "Migrating to Real Application Clusters from Single-Instance Oracle Databases"](#)

Migrating to Real Application Clusters from Single-Instance Oracle Databases

This chapter describes the procedures for migrating from single-**instance** Oracle databases to cluster-enabled **Real Application Clusters** databases. It also describes procedures for upgrading from Oracle8i Parallel Server to Oracle9i Real Application Clusters. The topics in this chapter are:

- [Migrating from a Single Instance to Real Application Clusters](#)
- [Migrating with Raw Devices or Shared File Systems](#)
- [Upgrading Oracle8i Configurations to Oracle9i Configurations](#)

Migrating from a Single Instance to Real Application Clusters

This section explains how to enable your single instance database to support Real Application Clusters. It also explains how to begin a project with a single-instance Oracle database and migrate to Real Application Clusters. This section covers the following topics:

- [Deciding to Migrate](#)
- [Preparing to Migrate](#)
- [Migrating Databases from Single Instance to Cluster-Enabled Environments](#)

See Also: *Oracle9i Database Migration* for complete procedures on migrating to Real Application Clusters

Deciding to Migrate

Do not migrate to Real Application Clusters if:

- You are not using a supported configuration of shared disks
- Your application was specifically designed to *not* use **cluster database** processing

If your platform supports a cluster file system, then you can use it for Real Application Clusters. You can also migrate to Real Application Clusters and use a non-shared file system. To do this, copy the `ORACLE_HOME` to the identical location on each **node** in your cluster.

Preparing to Migrate

To migrate to Real Application Clusters, consider both the hardware and software requirements and the administrative issues described in this section.

Hardware and Software Requirements

To migrate to Real Application Clusters you must have:

- A supported hardware and operating system software configuration
- A license for the Oracle9i Real Application Clusters

Administrative Issues of Migrating Applications from Single-Instance to Cluster-Enabled Environments

Note the following administrative issues of conversion:

- Backup procedures should be available before migrating from a single-instance Oracle database to Real Application Clusters.
- Additional archiving considerations apply in Real Application Clusters environments. In particular, the archive file format requires a thread number. Furthermore, archived logs from all nodes are needed for media recovery. If you archive to a file and you do not use a cluster file system, then a method of accessing the archive logs is required where file systems are not shared.

Migrating Databases from Single Instance to Cluster-Enabled Environments

This procedure explains how to migrate a database from single instance to cluster-enabled Oracle. This procedure assumes you are currently using a file system for your single instance Oracle. To use Real Application Clusters, you must migrate the database from the file system to **raw devices**.

If your database already uses a shared file system, review the additional notes about these procedures under the heading "[Migrating with Raw Devices or Shared File Systems](#)" on page 10-7.

To migrate a database from a single instance to Real Application Clusters, complete the following procedures:

[Task 1: Configure Hardware](#)

[Task 2: Create Raw Devices](#)

[Task 3: Evaluate Tablespaces and Log Files of Single Instance](#)

[Task 4: Export Data from Old Database](#)

[Task 5: Install Operating System-Dependent Cluster Software](#)

[Task 6: Install Oracle9i Enterprise Edition and Oracle9i Real Application Clusters](#)

[Task 7: Create the Database](#)

[Task 8: Import from Old Database into New Database](#)

[Task 9: Adjust Parameters](#)

[Task 10: Start the Database](#)

Task 1: Configure Hardware

See your vendor documentation for information about setting up Real Application Clusters hardware.

Task 2: Create Raw Devices

Create the raw devices needed for the datafiles, control files, and redo log files for Real Application Cluster databases as explained in "[Configuring Shared Disk Subsystems for Real Application Clusters](#)" on page 2-2.

Task 3: Evaluate Tablespaces and Log Files of Single Instance

Because each tablespace on the single instance database must have a matching, identical tablespace on the Real Application Clusters database, it is important that you consolidate, add, or rename the tablespaces on the single instance at this time if needed.

Each additional node in the cluster requires at least two redo log files. Typically, a single instance database only has two redo log files. Therefore, add redo log files for each node using the `ALTER DATABASE ADD LOGFILE` statement.

Note: If you are not changing operating system platforms for your database, then you do not need to perform [Task 4: Export Data from Old Database](#) and [Task 8: Import from Old Database into New Database](#). Instead, use the `dd` command in UNIX or the `OCOPY` command in Windows environments, to move files from the file system to the raw devices. Before executing this step, you must appropriately size the raw devices that correspond to each datafile.

Task 4: Export Data from Old Database

Export the entire database from the single instance database. Use a tool such as the Export utility by entering the following:

```
exp username/password file=file.dmp full=y log file.log
```

Where `file.dmp` represents the data from the full database export of the database, and `file.log` represents the log of the operation.

Note: To export an entire database, use the user name `SYSTEM`; do not use `SYS`.

See Also: *Oracle9i Database Utilities* for further information about this tool

Task 5: Install Operating System-Dependent Cluster Software

For UNIX, refer to your vendor's operating system-dependent documentation for instructions about installing operating system-dependent cluster software.

For Windows NT and Windows 2000, use the Oracle-supplied preinstall tool to install the Oracle OSD clusterware.

Task 6: Install Oracle9i Enterprise Edition and Oracle9i Real Application Clusters

Except on the server already running the **Oracle9i Enterprise Edition**, install Oracle9i Enterprise Edition along with the Oracle9i Real Application Clusters option as described in *Oracle9i Real Application Clusters Setup and Configuration*.

If your current single instance database does not have the supporting hardware, perform a clean install on new cluster hardware. On the server already running Oracle9i Enterprise Edition, install Oracle9i Real Application Clusters using one of the preconfigured database configuration types.

Task 7: Create the Database

Create a new database on the raw partitions. Depending on the install type and subsequent configuration options, you can create a database with the following methods:

- If you selected one of the preconfigured database configuration types or the **Customized** database configuration type, the **Database Configuration Assistant (DBCA)** creates the database after installation.
- If you requested that the DBCA not run during the installation, you can still run the DBCA to create a database, or create the database manually as described in *Oracle9i Real Application Clusters Setup and Configuration*.

Task 8: Import from Old Database into New Database

Import the entire database into the empty database using a tool such as the Import utility. To run the Import utility:

```
imp system/password file=file.dmp full=y log file.log
```

`file.dmp` represents the data from the full database export of the database, while `file.out` represents the log of the operation.

See Also: *Oracle9i Database Utilities* for more information about this tool

Note: If the original database from which the export file was generated contains tablespaces that are not in the new database, the Import utility attempts to create those tablespaces with associated datafiles. Resolve this by ensuring both databases contain the same tablespaces. The datafiles do not have to be identical. Only the tablespace names need to be the same.

Task 9: Adjust Parameters

You should make the following parameter changes to accommodate Real Application Clusters:

- Set the `INSTANCE_NUMBER` parameter to a unique value for each instance.
- If you optimized memory use on your single instance database, you need to adjust the size of the SGA to avoid swapping and paging when you migrate to Real Application Clusters. This is because Real Application Clusters requires about 350 bytes for each buffer too accommodate the **Global Cache Service (GCS)**. For example, if you have 10,000 buffers, Real Application Clusters requires about 350*10,000 bytes more memory. Therefore, adjust the size of the SGA to avoid swapping and paging by changing the `DB_CACHE_SIZE` and `DB_nK_CACHE_SIZE` parameters accordingly.

See Also: *Oracle9i Database Performance Tuning Guide and Reference* for more details about optimizing your memory configuration

Task 10: Start the Database

For more information on this procedure, refer to *Oracle9i Real Application Clusters Administration*.

Migrating with Raw Devices or Shared File Systems

If your database is already using raw devices on shared disk storage, or if you are using certain operating system file system architectures that are vendor-provided, then you do not have to rebuild the database as described in the previous procedures. Instead:

1. Perform steps 1 and 2, making sure to add the required raw devices for the log files.
2. Skip steps 3 and 4.
3. Perform steps 5 and 6.
4. Skip steps 7 and 8.
5. For step 9, start the additional instance.

The procedure in this case is much simpler than the procedure described in ["Migrating Databases from Single Instance to Cluster-Enabled Environments"](#) on page 10-3. The process may only require that you execute an operating system `copy` command to move your files to a new location, rather than having to rebuild the database.

See Also: *Oracle9i Database Migration* for procedures for upgrading from Oracle 7.3, Oracle 8.0, and Oracle8i to Oracle9i release 2 (9.2)

Upgrading Oracle8i Configurations to Oracle9i Configurations

If you are upgrading from Oracle8i Parallel Server to Oracle9i Real Application Clusters, upgrade your configuration information using the following post-installation procedure. Do this for each Real Application Clusters database:

1. Stop all **Global Services Daemon (GSD)** processes using the `gsdctl stop` command.
2. Execute the following command from the node in your cluster on which the `db_name.conf` file is located:

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

The environment variable referred to in this example is an Oracle8i environment variable.

Note : Environment variables are case-sensitive.

Part V

Reference for Real Application Clusters Setup and Configuration

Part V provides reference information for Real Application Clusters. The contents of Part V are:

- [Appendix A, "Directory Structure for Real Application Clusters Environments"](#)
- [Appendix B, "Troubleshooting Oracle Enterprise Manager in Real Application Clusters"](#)
- [Glossary](#)

A

Directory Structure for Real Application Clusters Environments

Appendix A describes the directory structure for **Real Application Clusters** software environments. Specific topics covered in this appendix are:

- [Understanding the Real Application Clusters Directory Structure](#)
- [UNIX Directory Structures for Real Application Clusters](#)
- [Windows Directory Structures for Real Application Clusters](#)
- [Shared Oracle Home](#)

Understanding the Real Application Clusters Directory Structure

When you install **Oracle9i Enterprise Edition** and Oracle9i Real Application Clusters, all subdirectories are under a top-level `ORACLE_BASE`. `ORACLE_HOME` and `admin` directories are also located under `ORACLE_BASE`.

UNIX Directory Structures for Real Application Clusters

Table A-1 shows the hierarchical directory tree of a sample OFA-compliant database for Real Application Clusters on UNIX platforms:

Table A-1 *Directory Structure for A Sample OFA-Compliant UNIX Environment*

Root	Second-Level	Third-Level	Fourth-Level	Fifth-Level
<code>\$ORACLE_BASE</code>				<code>/u01/app/oracle</code> The default <code>ORACLE_BASE</code> directory
	<code>\$ORACLE_HOME</code>			<code>/product/9.2</code> The name of the Oracle home by default
		<code>/bin</code>		Subtree for Oracle binaries
		<code>/network</code>		Subtree for Oracle Net

See Also: *Oracle9i Administrator's Reference* for your UNIX operating system for further information about the `$ORACLE_HOME` and `/admin` directories

Windows Directory Structures for Real Application Clusters

Table A-2 shows the hierarchical directory tree of a sample OFA-compliant database for Real Application Clusters on Windows NT and Windows 2000:

Table A-2 Directory Structure for A Sample OFA-Compliant Windows Environment

Root	Second-Level	Third-Level	Fourth-Level	Fifth-Level
x:\oracle_base				c:\oracle The default ORACLE_BASE directory.
	\%ORACLE_HOME%			\ora9.2 The name of the Oracle home by default.
		\bin		Subtree for Oracle binaries.
		\network		Subtree for Oracle Net configuration files, including <code>tnsnames.ora</code> , <code>listener.ora</code> and <code>sqlnet.ora</code> .
		\srvm		\admin subdirectory.
			\admin	The Real Application Clusters script <code>clustdb.sql</code> and initialization parameter files for database creation
		\database		This is a legacy directory from previous releases. It contains initialization files that point to the new directory location for the initialization parameter files, <code>ORACLE_BASE\admin\db_name\pfile</code> .
	\admin			Subtree for Real Application Clusters database administration files
		\db_name		<code>db_name</code> database administration files for the instance identified by <code>sid</code> .

Table A-2 (Cont.) Directory Structure for A Sample OFA-Compliant Windows

Root	Second-Level	Third-Level	Fourth-Level	Fifth-Level
			\ad hoc	Ad hoc SQL scripts.
			\adump	Audit files.
			\arch	Archived redo log files.
			\bdump	Background process trace files.
			\cdump	Core dump files.
			\create	Programs used to create the database.
			\exp	Database export files
			\pfile	Initialization parameter files
			\udump	User SQL trace files

Shared Oracle Home

If the Oracle home location is shared (through NFS or any other equivalent facility) by each **node**, set up **Oracle Intelligent Agent** as follows:

1. Install Oracle Intelligent Agent for each node in its own Oracle home location, distinct from the shared Oracle home location.

Note: You cannot install the Intelligent Agent in the same shared Oracle home location where your Oracle database is located.

2. Copy or link `tnsnames.ora` and `listener.ora` to each Oracle Intelligent Agent's Oracle home location from the shared Oracle home.
3. Before starting Oracle Intelligent Agent on each node, set the Oracle home location to Oracle Intelligent Agent's Oracle home.
4. Issue the `agentctl start` command. Ensure that the Agent, or the DBSNMP utility, is set up to run at system startup time.

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

B

Troubleshooting Oracle Enterprise Manager in Real Application Clusters

Appendix B describes **service discovery** issues for **Oracle Enterprise Manager** in **Real Application Clusters** environments. Specific topics covered in this appendix are:

- [Resolving Service Discovery Failures](#)
- [Understanding Discovery](#)
- [Discovery Results](#)
- [Troubleshooting Discovery](#)

Resolving Service Discovery Failures

Discovery of nodes and objects by Oracle Enterprise Manager is robust and rarely fails once a correct configuration is established. Failures typically occur because:

- The Oracle Intelligent Agent is not started on the [node](#)
- The Global Services Daemon (GSD) is not running
- The configuration is incorrect

If starting the Oracle Intelligent Agent and the GSD does not resolve the problem, then the discovery failure could be due to a more serious configuration issue.

Understanding Discovery

During discovery, a `services.ora` file on the managed nodes is created in the `$ORACLE_HOME/network/agent` directory on UNIX operating systems and `%ORACLE_HOME%\network\admin` directory on Windows NT or Windows 2000 operating systems. This file contains information about the nodes and their services (databases, instances, and listeners) discovered.

If you used the Database Configuration Assistant to create your database, then the appropriate discovery information will be configured for you automatically. However, if you manually configure your database, then you must manually create the entries for this configuration information as described in the following sections.

Note: Oracle8*i* and Oracle9*i* databases can co-exist on the same node. Refer to Oracle8*i* documentation for information on Oracle8*i* Discovery.

The `services.ora` file is created from the Real Application Clusters configuration information in the shared raw device (or shared cluster file system file maintained by the [SRVCTL Utility](#)) and the following sources on the managed nodes:

- [The UNIX oratab File and the Windows NT and Windows 2000 Registry](#)
- [SRVM Configuration Repository](#)
- [The listener.ora File](#)
- [The tnsnames.ora File](#)
- [The sqlnet.ora File](#)

You must accurately configure each of these components so that discovery succeeds.

Note: The following Discovery descriptions apply only to Release 1 (9.0.1) and later. If you are running an earlier version, please refer to the corresponding editions of the Oracle Parallel Server documentation.

The UNIX oratab File and the Windows NT and Windows 2000 Registry

Discovery first determines the Real Application Clusters database name and the nodes associated with the database. How the discovery process accomplishes this is operating-system dependent.

- [UNIX and oratab](#)
- [Windows NT and Windows 2000](#)

UNIX and oratab

On UNIX operating systems, discovery uses information in the `oratab` entry for the name of the cluster database. The location of the file for the `oratab` entry is platform specific and is found in either the `/etc/oratab` or `/var/opt/oracle/oratab` directories. This file contains entries of the form:

```
db_name:$ORACLE_HOME:N
```

Where `db_name` is the database name and `$ORACLE_HOME` is the Oracle home given to your database. From this entry, the database name is acquired.

Next, discovery executes the following command to retrieve the instances-to-node mappings of the Real Application Clusters database:

```
srvctl config database -d db_name
```

Where `db_name` was retrieved from `oratab`.

Windows NT and Windows 2000

Obtain the value of `INSTANCE_SID` from the following Registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOMEID\ORACLE_SID
```

SRVM Configuration Repository

Discovery requires that the configuration for each instance be recorded in the SRVM repository. To determine whether the configuration for an instance is recorded, enter the following command:

```
srvctl config database -p db_name
```

If the output from this command reveals that entries are missing from the repository, then manually add the configuration for the affected instances and databases as described in "[Task 3: Create the Real Application Clusters Configuration with SRVCTL](#)" on page 5-12.

The listener.ora File

Discovery locates the **listener** and **instance** names for a node from the **listener.ora** file located in `$ORACLE_HOME/network/admin` on UNIX operating systems and `%ORACLE_HOME%\network\admin` on Windows NT or Windows 2000 operating systems on the discovered nodes.

Discovery requires the following entries:

- The listener address must contain a TCP/IP address that specifies the `HOST` value as the host name of the discovered node. This ensures that the listener actually resides on the node.

```
(description=
  (address=(protocol=tcp)(host=db1-server)(port=1521)))
```

- For each listener that runs on the node, the `sid_LIST_listener_name` entry is searched for a description (`sid_DESC`) that contains the instance name (`sid_NAME`):

```
sid_list_listener=
  (sid_list=
    (sid_desc=
      (sid_name=db1)))
```

The `listener.ora` file created after installation typically contains the configuration for discovery.

See Also: *Oracle9i Real Application Clusters Administration* for more information on the `listener.ora` file

How SRVCTL Operations Associate Listeners with Instances

When SRVCTL operations attempt to start instances and their associated listeners, the listener that SRVCTL will start is determined from the static configuration information set for the `SID_NAME` parameter in the `SID_LIST_LISTENER` entry in the `listener.ora` file. For example, the `SID_NAME` is set for the instance a the `listener.ora` file as follows:

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME = /ora/home)
      (SID_NAME = instance1)
    )
  )
)
```

For this example, when SRVCTL starts `instance1` SRVCTL will also attempt to start the listener named `LISTENER`.

The tnsnames.ora File

The discovery process determines the service names and connect descriptors information for the database and its instances using the entries in the `tnsnames.ora` file. This file is located in `$ORACLE_HOME/network/admin` on UNIX operating systems and `%ORACLE_HOME%\network\admin` on Windows NT or Windows 2000 on the discovered nodes, is read by the discovery process to determine names and address information for the database and instances on a node.

Discovery requires the following entries:

- Each instance must have an entry in `tnsnames.ora` file For example:

```
db1.us.acme.com=
  (description=
    (address=(protocol=tcp)(host=db1-server)(port=1521))
    (connect_data=
      (service_name=db.us.acme.com)
      (instance_name=db1)))
```

- The Real Application Clusters database should have an entry. For example:

```
db.us.acme.com=
  (description=
    (load_balance=on)
    (failover=on)
    (address=(protocol=tcp)(host=db1-server)(port=1521)))
```

```
(address=(protocol=tcp)(host=db2-server)(port=1521))
(connect_data=
  (service_name=db.us.acme.com)))
```

The sqlnet.ora File

The **sqlnet.ora** file is automatically configured with:

- The computer's domain which is automatically appended to any unqualified net service name or service name. For example, if the default domain is set to `us.acme.com`, Oracle resolves `db` in the connect string `CONNECT scott/tiger@db as: db.us.acme.com`.
- A **naming method** the server can use to resolve a name to a connect descriptor. The order of naming methods is as follows: directory naming (for the Customized installations only), `tnsnames.ora` file, Oracle Names server, and host naming.

The following is a sample `SQLNET.ORA` file created during a preconfigured database configuration install:

```
names.default_domain=us.acme.com
names.directory_path=(tnsnames, onames,hostname)
```

See Also: *The Oracle9i Net Services Administrator's Guide* for further information about the `sqlnet.ora` file

Discovery Results

Discovery results in the creation of:

- Discovered nodes and services listed in `services.ora` file in the `$ORACLE_HOME/network/agent` directory on UNIX operating systems and `%ORACLE_HOME%\network\admin` directory on Windows NT or Windows 2000.

The `services.ora` file should contain an `ops_database` entry for the Real Application Clusters database, that lists the node, database address, and name of the database. The following example shows a database named `db.us.acme.com` running on node `db1-server` and `db2-server`. The database address comes from the `db.us.acme.com` **net service name** in the `tnsnames.ora` file.


```
db.us.acme.com=(ops_database, db1-server, (description=(load_
balance=on)(failover=on) (address=(protocol=tcp)(host=db1-server)
(port=1521)) (address=(protocol=tcp)(host=db2-server)(port=1521))(connect_
data=(service_name=db.us.acme.com))), db.us.acme.com)
```

The `services.ora` file should also contain an `OPS_INSTANCE` entry for the instance that runs on the node. This entry identifies the:

- Name of the node
- Net service entry for the instance from the `tnsnames.ora` file
- Real Application Clusters database name to which the instance belongs
- Listener name

The following example shows the instance `db1` runs on node `db1-server`, and is listened for by `listener_db1-server`:

```
db1.us.acme.com=(ops_instance, db1-server,
(description=(address=(protocol=tcp)(host=db1-server)(port=1521))(connect_
data= (service_name=db.us.acme.com)(instance_name=db1))), db.us.acme.com,
(PROP=(LSNRS=listener_db1-server)(ORACLE_HOME=/private/system/db:n)(ORACLE_
SID=db1))))
```

- Errors logged in `nmiconf.log` in `$ORACLE_HOME/network/log` directory on UNIX operating systems and `%ORACLE_HOME%\network\log` directory on Windows NT or Windows 2000.

Troubleshooting Discovery

Note: The Oracle Intelligent Agent must be running and the Global Services Daemon (GSD) must be started before Discovery will succeed. Also note that the Agent must be running in an Oracle Home in which you installed Oracle Real Application Clusters.

If the `services.ora` file contains an `ORACLE_DATABASE` entry instead of the `ops_database` and `ops_instance` entries, then discovery has failed. To resolve this:

1. Check that the database is configured correctly:

On UNIX:

- Verify that `oratab` file is configured correctly.

On Windows NT or Windows 2000:

- Check the registry entries associated with the database.

2. Run the following command to verify the configuration:

```
SRVCTL config database -d db_name
```

SRVCTL displays the name of the node and the instance for the node. The following example shows a node named `db1-server` running an instance named `db1` with a listener named `LISTENER_NODE1`. If you execute the following command:

```
srvctl config database -d db
```

The output should appear similar to:

```
db1-server db1 /oracle
db2-server db2 /oracle
```

3. Inspect the `listener.ora` and `tnsnames.ora` file entries to ensure that the required entries are present.

See Also: *Oracle9i Real Application Clusters Administration* for information on troubleshooting with Oracle Enterprise Manager Thread Trace Files, and how to contact Oracle Support Services

Glossary

administrative context

A directory entry under which an Oracle Context (cn=OracleContext) resides. During directory access configuration, clients are configured with an administrative context in the directory configuration file, ldap.ora. The administrative context specifies the location of the Oracle Context in the directory whose entries a client expects to access.

automatic undo management

A feature that automatically manages the sizing of undo tablespaces.

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.

client load balancing

Load balancing, whereby if more than one listener services a single database, a client can randomly choose between the listeners for its connect requests. This randomization enables all listeners to share the burden of servicing incoming connect requests.

cluster

A set of instances that cooperates to perform the same task.

cluster database

The generic term for 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 cluster membership across the [cluster](#). The CM also monitors process status, specifically the status of the database instance. The [Global Services Daemon \(GSD\)](#), a background process that monitors the status of the [Global Cache Service \(GCS\)](#), registers and de-registers from the 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 identifier

A [net service name](#) or [service name](#), that resolves to a [connect descriptor](#).

connection load balancing

A feature that balances the number of active connections among various instances and [shared server dispatchers](#) for the same service.

connect-time failover

A client connect request is forwarded to a another listener if first listener is not responding. Connect-time failover is enabled by [service registration](#), because the listener knows if an instance is up prior to attempting a connection.

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.

Customized

The Database Configuration Assistant's Customized configuration type enables you to use the DBCA to create a customized database. Select the Customized installation to create customized database objects.

Database Configuration Assistant (DBCA)

A GUI tool for creating a database for an online transaction processing (OLTP), data warehouse, or **hybrid** database environment. The DBCA creates a standard set of tablespaces for the type of database you select.

The DBCA also has administrative features such as instance and template management. Use the instance management feature to add or delete instances. Use the template management feature to manage and customize your database creation scripts. You can also use template management to reverse engineer databases to preserve the attributes of existing databases for which you do not have creation scripts.

Database Upgrade Assistant (DUA)

A GUI tool for upgrading a database from one release of Oracle to another. The DUA fully supports upgrades of Oracle cluster-enabled software from one release to another.

Data Warehouse

The Database Configuration Assistant's preconfigured database template for a data warehouse environment. This template includes datafiles.

dedicated server

A server that requires a dedicated server process for each user process. There is one server process for each client.

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**.

Enterprise Manager Configuration Assistant (EMCA)

A tool for creating, deleting, and modifying Oracle Enterprise Manager configurations and settings.

extended partition

A type of partition on Windows NT and Windows 2000 that points to raw space on the disk. An extended partition can be assigned multiple logical drives to accommodate datafiles, control files, and redo log files.

external procedures

Functions or procedures written in a third-generation language (3GL) that can be called from PL/SQL code. Only C is supported for external procedures.

General Purpose

The Database Configuration Assistant's preconfigured database template for a hybrid database environment. This template includes datafiles.

global cache parameters

Initialization parameters determine the size of the collection of global resources that protect the database buffers on all instances. These parameters should be set in your parameter file.

Global Cache Service (GCS)

The process that implements **Cache Fusion**. It maintains the block mode for blocks in the global role. It is responsible for block transfers between instances. The Global Cache Service employs various background processes such as the **Global Cache Service Processes (LMSn)** and **Global Services Daemon (GSD)**.

Global Cache Service Processes (LMSn)

The processes that handle remote **Global Cache Service (GCS)** messages. **Real Application Clusters** provides for up to 10 Global Cache Service Processes. The number of LMSn varies depending on the amount of messaging traffic for each **node** in the cluster.

global database name

The full name of the database that uniquely identifies the database from any other database. The global database name is of the form `database_name.database_domain`, for example, `sales.us.acme.com`.

Global Services Daemon (GSD)

A process that receives requests from the **SRVCTL Utility** to execute administrative tasks, such as startup or shutdown. The command is executed locally on each node and the results are returned to SRVCTL. The GSD is installed on all nodes.

hybrid

A database that has both online transaction and data warehouse processing characteristics.

initialization parameter file

Files that contains information to initialize the database (**initdbname.ora**) and instances (**initsid.ora**).

initsid.ora

An instance initialization parameter file that contains parameters unique for an **instance** and that points to **initdbname.ora** for database parameters.

initdbname.ora

An initialization parameter file that contains global parameters that apply to an entire cluster.

instance

The combination of the System Global Area (SGA) and each process for the Oracle database. The memory and processes of an instance manage the associated database's data and serve the database users. Each instance has a unique **Oracle system identifier (sid)**, **INSTANCE_NAME**, **INSTANCE_NUMBER**, rollback segments, and **thread number**.

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)**.

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.

Inter-process Communication (IPC)

An operating system-dependent component that transfers of messages and consistent-read versions of data blocks between instances on different nodes.

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.

LDAP

See [Lightweight Directory Access Protocol \(LDAP\)](#).

Lightweight Directory Access Protocol (LDAP)

A protocol for accessing on-line directory services.

Lock Manager Servers (LMSn)

See [Global Cache Service Processes \(LMSn\)](#).

LMD process

A process that handles remote resource requests. Remote requests are requests that originate from another instance.

LMON

See [Global Services Daemon \(GSD\)](#).

Management Server

The [Oracle Enterprise Manager](#) Management Server provides centralized intelligence and distributed control between the [Console](#) and the managed nodes. It also 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, which is a set of tables stored in a database.

multiple Oracle homes

The capability of having more than one Oracle home directory on a machine.

naming method

The method that client applications use to resolve a [net service name](#) to a [connect descriptor](#).

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 with a net service name in a connect string for the service to which they wish to connect.

New Database

The Database Configuration Assistant's template that does not include datafiles. This template is fully customizable.

node

A machine on which an instance resides.

Object Link Manager (OLM)

A component of the Cluster Setup Wizard that you use to create persistent symbolic links to the logical drives required by Oracle Real Application Clusters. The Object Link Manager is associated with the [Oracle Object Service](#).

operating system-dependent (OSD) clusterware

Software clusterware tailored for various operating systems. OSD clusterware provides communication links between the operating system and the [Real Application Clusters](#) software.

Oracle9i Enterprise Edition

Oracle9i Enterprise Edition is an Object-Relational Database Management System (ORDBMS). It provides the applications and files to manage a database. All other Oracle9i Real Application Clusters components are layered on top of Oracle9i Enterprise Edition.

Oracle Object Service

This service updates all nodes when symbolic links are modified. When installed, this service is set to autostart, so that it starts whenever you restart your system.

optimal flexible architecture (OFA)

A set of file naming and placement guidelines for Oracle software and databases.

Oracle Context

An entry in a LDAP-compliant directory of `cn=OracleContext`, under which all Oracle software relevant information is kept.

Oracle Data Gatherer

The Oracle Data Gatherer collects performance statistics for the [Oracle Performance Manager](#). The Oracle Data Gatherer must be installed on a node somewhere on the network.

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 Net Manager

A graphical user interface tool that combines configuration abilities with component control to provide an integrated environment for configuring and managing **Oracle Net**. It can be used on either the client or server.

Oracle Net Configuration Assistant

A post-installation tool that configures basic network components after installation, such as the listener names, protocol addresses, and naming methods the client will use to resolve each **connect identifier**.

Oracle Managed Files

An automated file system that controls the use of files within Oracle. Oracle Managed Files requires a clustered file system.

Oracle Performance Manager

Part of **Oracle Enterprise Manager** that offers tabular and graphic performance statistics charts for single-instance Oracle databases and Real Application Clusters.

Oracle Real Application Clusters

Oracle Real Application Clusters is a breakthrough 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 services

Oracle services are created and associated with Oracle products, such as the database or listener.

Oracle system identifier (*sid*)

A name that identifies a specific instance of a running pre-release 8.1 Oracle database. For an Oracle9i 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 number** 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.

Oracle Universal Installer (OUI)

A graphical user interface tool that facilitates the installation of the Oracle database software and its related components.

parallel execution

Divides the work of processing certain types of SQL statements among multiple parallel execution server processes. Commonly used in **Data Warehouse** applications.

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.

quorum disk

See **voting disk**.

raw device

A disk or partition on a disk drive that does not have a file system set up on it.

raw partition

The raw, unformatted devices on shared disk arrays as used in Windows NT and Windows 2000 platforms.

Real Application Clusters

See [Oracle Real Application Clusters](#).

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 (Recovery Manager)

An Oracle tool that backs up, copies, restores, and recovers datafiles, control files, and archived redo logs.

seed database

A preconfigured, ready-to-use database that requires minimal user input to create.

server clustering

See [Real Application Clusters](#).

Server Management (SRVM)

A comprehensive and integrated system management solution for Real Application Clusters. Use Server Management to manage cluster databases running in heterogeneous environments through an open client/server architecture through [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 big-picture view of all nodes and their respective services.

service name

A logical representation of a database, which 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 by which the PMON process 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 server

A server that is configured to allow many user processes to share very few server processes. This increased the number of users that can be supported. With shared server configurations, many user processes connect to a **dispatcher**. Contrast this with **dedicated server**.

sid

A *sid* is an abbreviation for **Oracle system identifier (sid)**.

Software Only

The Oracle Universal Installer's database configuration type that only copies the software onto your hard drive. This configuration type does not use datafiles or perform configuration tasks.

spfile.ora

The binary parameter file that resides on the Oracle Server.

sqlnet.ora

A configuration file for the client or server that specifies the client domain to append to unqualified service names or net service names, the order of naming methods the client should use when resolving a name, the logging and tracing features to use, and other network information.

SRVCTL Utility

A utility for administering instances and databases. SRVCTL gathers information about all the instances for **Oracle Enterprise Manager**. 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 to SRVCTL. SRVCTL on that node then communicates to the other nodes through **Oracle Net**.

starter database

A preconfigured, ready-to-use database that requires minimal user input to create.

startup (START)

An operating system-dependent component that provides one-time configuration to startup functionality.

symbolic link name

A name for a Windows NT and Windows 2000 logical partition.

SYSDBA

A database administration role that contains all system privileges with the ADMIN OPTION and the **SYSOPER** system privileges. SYSDBA also permits CREATE DATABASE and time-based recovery.

SYSOPER

A database administration role that enables a database administrator to perform STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG, and RECOVER, and includes the RESTRICTED SESSION privilege.

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

A configuration file that contains **net service names** mapped to **connect descriptors**. The tnsnames.ora file typically resides in \$ORACLE_HOME/network/admin on UNIX platforms and %ORACLE_HOME%\network\admin on Windows NT and Windows 2000 platforms. This file is needed on clients, nodes, the **Console**, and on the **Oracle Performance Manager** machine.

Transaction Processing

The Database Configuration Assistant's preconfigured database template for a transaction processing environment. This template includes datafiles.

transparent application failover (TAF)

A runtime failover for high-availability environments, such as Real Application Clusters and Oracle Real Application Clusters Guard, that 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.

Transmission Control Protocol/Interconnect Protocol (TCP/IP)

A set of protocols that govern how information is packaged and transferred across the internet to guarantee reliable service.

User Datagram Protocol (UDP)

A transport layer protocol defined by the US Department of Defense (DoD) for use with the IP network layer. It provides a best-effort datagram service to an End System. The service provided by UDP is an unreliable service because it does not have a delivery guarantee or protection from duplication.

Virtual Interface Architecture (VIA)

An industry-standard architecture for inter-cluster communications. VIA's rapid server-to-server communications enhances an application's scalability and performance. VIA does this by allowing a single application to run efficiently across dozens of clustered nodes and by accelerating the data exchange among distributed application modules running on different application servers.

voting disk

Real Application Clusters uses a voting disk to improve cluster availability. Oracle stores cluster status information on the partition reserved for the voting disk.

warming the library cache

The process of transferring the most important information about parsed SQL statements and compiled PL/SQL units from the library cache on the primary instance to the library cache on the secondary instance. Warming the cache improves performance after failover because the library cache is already populated.

Index

A

- acquiring rollback segments, 5-6
- ACTIVE_INSTANCE_COUNT parameter, 9-6
- ADD LOGFILE clause
 - thread required, 5-8
- adding a file, 5-9
- All Initialization Parameters dialog, 4-13
- allocation
 - rollback segments, 5-6
- ALTER DATABASE DISABLE THREAD
 - statement, 5-8
- ALTER DATABASE statement
 - DISABLE clause, 5-8
 - THREAD clause, 5-8
 - thread of redo, 5-8
- ALTER ROLLBACK SEGMENT statement, 5-6
- architecture
 - Optimal Flexible Architecture (OFA), 3-8, 4-2
 - Oracle Enterprise Manager, 7-5
 - SRVCTL Utility on UNIX, 7-3
- archive logs
 - destinations, conversion to multi-instance, 10-3
- ARCHIVELOG mode, 5-3
- ASCII file
 - for UNIX raw volume names, 2-10
- automatic undo management, 5-5, 8-7
 - rollback segments, 5-4

B

- backups
 - after creating new databases, 5-18
 - before database creation, 5-10

- conversion to multi-instance, 10-3
- server parameter file, 6-7

C

- client load balancing
 - configuring, 9-10
 - description of, 9-8, 9-10
- clustdb.sql script, 5-17
- cluster database
 - reasons not to convert to, 10-2
- clustered file systems
 - as required by Oracle Managed Files, 1-6
- clusterware
 - diagnostics, 3-4
- Communication Daemon, defined, 7-2
- components, 9-12
 - created when using the DBCA, 8-4
- concurrency
 - maximum number of instances, 5-2
- Configuration Assistant, 7-8
- configuration file, 8-2
- configuration raw device, 2-2
- configuration types
 - Customized, 3-5, 3-6
 - Data Warehouse, 3-5
 - General Purpose, 3-5
 - New Database, 3-6
 - Software Only, 3-6
 - Transaction Processing, 3-5
- configurations
 - change in redo log, 5-9
- configuring
 - clients

- client load balancing, 9-10
- connect-time failover, 9-10
 - primary and secondary instances, 9-7
- global database name, 4-9
- high availability, 9-1 to 9-12
 - primary and secondary instances, 9-6
- listeners
 - primary and secondary instances, 9-8
- Oracle Enterprise Manager, 7-10, 7-11
- Oracle Performance Manager, 7-11
- primary and secondary instances, 9-6
- Server Management, 7-1
- shared disks, 2-1
- SID Prefix, 3-9, 3-10
- connecting
 - Real Application Clusters database, 9-12
- connection load balancing, 8-8
- connect-time failover
 - configuring, 9-10
 - described, 9-8, 9-10
 - GLOBAL_DBNAME parameter in listener.ora file, 8-11
- Console
 - setup, 7-5
- control files, 1-5, 2-2
 - creating, 5-4
 - described, 8-7
 - log history, 5-2
 - raw devices, 2-6
- CREATE CONTROLFILE statement, 5-4
 - changing database options, 5-4
- CREATE DATABASE, 5-2
 - creating threads, 5-7
 - MAXDATAFILES clause, 5-3
 - MAXINSTANCES clause, 5-2
 - MAXLOGFILES clause, 5-2
 - MAXLOGHISTORY clause, 5-2
 - MAXLOGMEMBERS clause, 5-2
- Create Database
 - option on Creation Options page, 4-17
- CREATE PUBLIC ROLLBACK SEGMENT
 - statement, 5-6
- CREATE ROLLBACK SEGMENT statement, 5-6
 - creating
 - initdb_name.ora file, 5-15

- initialization parameter files, 5-15
- init sid.ora file, 5-15
- new redo log groups, 5-8
- Oracle Enterprise Manager repository, 7-8
- password files, 5-16
- raw devices on UNIX, 2-7
- raw devices on Windows, 2-12
- Real Application Clusters database
 - backing up the new database, 5-18
 - initdb_name.ora file, 5-15
 - init sid.ora file, 5-15
 - ORACLE_SID environment variable, 5-13
 - OracleServicesid service, 5-14
 - oratab file, 5-13, 8-3
 - password files, 5-16
 - with the Database Configuration Assistant, 3-8, 4-2, 4-3
- Real Application Clusters database with manual methods, 5-9
- rollback segments, 5-5, 5-6
- server parameter files, 6-7
- threads, 5-7
- Creation Options page, 4-17
- cross-node registration, 8-8
- crtsrv batch file, 5-14
- Customized
 - configuration type, 3-5, 3-6, 3-7
- CWMLITE tablespace
 - raw devices, 2-6
- cwmlite1
 - identifying on UNIX, 2-9

D

- data dictionary
 - views, 5-6
- Data Warehouse
 - configuration type, 3-5, 3-6
- database
 - archiving mode, 5-8
 - components, created when using the DBCA, 8-4
 - configurations, types, 3-6
 - number of instances, 5-2
 - rollback segments, 5-5
- Database Configuration Assistant

- All Initialization Parameters dialog, 4-13
- components created by, 8-4
- control files, 8-7
- creating Real Application Clusters database
 - after installation, 4-3
 - during installation, 3-8, 4-2
- Creation Options page, 4-17
- Database Connection Options page, 4-11
- Database Features page, 4-10
- Database Identification page, 4-9
- Database Storage page, 4-15
- Database Templates page, 4-8
- datafiles, 8-4
- deleting databases with, 4-19
- deleting Real Application Clusters databases, 4-19
- initialization parameter files, 8-7
- Initialization Parameters page, 4-12
- List of Cluster Databases page, 4-21
- Node Selection page, 4-7
- Operations page, 4-6, 4-20
- processing during installation, 3-8
- raw devices for, 2-3
- redo log files, 8-6
- rollback segments, 8-7
- Summary dialog, 4-18, 4-22
- tablespaces, 8-4
- troubleshooting, 4-5
- using, 4-2
- Welcome page, 4-4
- database configuration type
 - selecting, 3-5
- database configuration types, 3-7
- Database Connection Options page, 4-11
- database directory, A-3
- Database Features page, 4-10
- Database Identification page, 4-9
- Database Storage page, 4-15
- Database Templates page, 4-8
- datafiles, 1-5, 2-2
 - and the DBCA, 8-4
 - described, 8-4
 - raw devices, 2-5
- db_name.conf file
 - service discovery, B-3

- DBA Studio
 - limitations, 7-7
- DBA_ROLLBACK_SEGS view
 - public rollback segments, 5-6
- DBCA_RAW_CONFIG
 - environment variable, 3-2
- DBMS_LIBCACHE package, 9-9
- dedicated servers, 8-8
- deleting databases
 - with the DBCA, 4-19
- devices
 - minimum configuration requirements, 2-2
- diagnostics
 - clusterware, 3-4
- directory structure, A-1, B-1
 - UNIX, A-2
- DISABLE THREAD clause, 5-8
- disabling
 - the archive history, 5-3
 - threads, 5-8
- dropping redo log files
 - manual archiving, 5-8
 - restrictions, 5-9
- DRSYS tablespace
 - description, 8-4
 - raw device, 2-6

E

- Edit User Preferences dialog box, 7-10
- environment variable
 - for raw devices, 2-10
 - UNIX, 3-2
- errors
 - messages, rollback segment, 5-6
 - Oracle Enterprise Manager discovery, B-2
- EXAMPLE tablespace
 - raw devices, 2-6
- example1
 - identifying on UNIX, 2-9
- exclusive mode
 - media recovery, 5-2
- Export utility, 10-4
- extent management
 - rollback segments, 5-5

F

- failover
 - and service registration, 8-8
- FAILOVER parameter, 9-11
- FAILOVER_MODE parameter, 9-2
 - for Transparent Application Failover, 9-2
- features, new, xxvii
- files
 - dropping, 5-8, 5-9
 - renaming, 5-9

G

- General Purpose
 - configuration type, 3-5
 - description of configuration type, 3-6
- Generate Database Creation Scripts
 - on Creation Options page, 4-17
- global database name, 4-9
- Global Services Daemon (GSD), 5-13
- global VS view tables, 7-2
- GLOBAL_DBNAME parameter, 8-11
- groups
 - MAXLOGFILES, 5-2
 - redo log files, 5-2, 5-9
- GSD, defined, 7-2

H

- hardware requirements, 1-6
- high availability
 - configuring, 9-1 to 9-12
 - primary and secondary instances, 9-6
 - Transparent Application Failover, 9-2
- host.equiv file, 2-11

I

- Import utility, 10-5
- INDX tablespace
 - described, 8-4
 - raw device, 2-6
- initdb_name.ora file
 - ACTIVE_INSTANCE_COUNT parameter, 9-6
 - creating, 5-15

- initialization parameter files, 8-7
 - creating for database creation, 5-15
 - editing before database creation, 5-15
 - for instances, 6-2
 - listener parameters, 8-9
- initialization parameters
 - DISPATCHERS, 8-8
 - LOCAL_LISTENER, 8-8
 - REMOTE_LISTENER, 8-8
- Initialization Parameters page, 4-12
- initsid.ora file
 - creating, 5-15
- installation
 - directory structure, A-1, B-1
 - ldap.ora file, 8-12
 - listener.ora file, 8-10
 - overview, 1-3, 1-5
 - preinstallation, 2-12
 - processing of the DBCA, 3-8
 - repository database, 7-8
 - requirements, hardware, 1-5, 1-6
 - requirements, hardware requirements, 1-6
 - requirements, software, 1-5
 - software requirements, 1-6
 - tnsnames.ora file, 8-12
 - verifying raw devices, 4-3
- installing
 - Oracle Enterprise Manager, 7-8
 - Oracle software with Oracle Universal Installer, 3-2
- INSTANCE_ROLE
 - for Transparent Application Failover, 9-2
 - use of in secondary instance connections, 9-8
- instances
 - adding instances, 5-2
 - initialization parameter files, 6-2
 - maximum number, 5-2
 - recovery, 5-2
 - recovery, starting another instance, 5-2
 - SID Prefix, 3-9, 3-10
 - thread number, 5-7
- Instances Logon dialog box, 7-12
- interconnects
 - supported for Real Application Clusters, 1-7

L

- ldap.ora file, 8-12
 - creating, 8-12
 - default configuration, 8-12
- library cache
 - warming, 9-9
- Lightweight Directory Access Protocol (LDAP), 8-12
- List of Cluster Databases page, 4-21
- listener
 - local, 8-8
 - registration, 8-11
- listener.ora file, 8-10
 - configuring, 8-9
 - default configuration, 8-10
 - Oracle Enterprise Manager service discovery, B-4
 - Server Management requirements, 7-4
 - sid_DESC parameter, B-4
 - sid_LIST_listener_name parameter, 5-19, 7-4, B-4
- listeners
 - GLOBAL_DBNAME parameter, 8-11
 - listener.ora file, 8-10
- load balancing
 - and service registration, 8-8
- LOAD_BALANCE parameter, 9-11
- local listener, 8-8
- log history, 5-2
- log switches
 - adding or dropping files, 5-9
- lsnodes
 - command syntax, 3-4

M

- manual archiving
 - dropping a redo log file, 5-8
- manual undo management, 5-6
- MAXDATAFILES clause, 5-3, 5-4
- MAXDATAFILES parameter, 5-4
- MAXINSTANCES clause, 5-2
 - changing, 5-4
- MAXINSTANCES parameter, 5-4

- MAXLOGFILES clause, 5-2, 5-4
- MAXLOGFILES parameter, 5-4
- MAXLOGHISTORY clause, 5-2
 - changing, 5-4
- MAXLOGHISTORY parameter, 5-4
- MAXLOGMEMBERS clause, 5-2, 5-4
- MAXLOGMEMBERS parameter, 5-4
- media recovery
 - log history, 5-2
- member
 - MAXLOGMEMBERS, 5-2
- migrating
 - Real Application Clusters issues, 4-23
 - single instance to Real Application Clusters, 10-2
 - to Real Application Clusters with raw devices, 10-7
 - to Real Application Clusters with shared file systems, 10-7
- multiple Oracle homes
 - UNIX, 4-23
 - Windows, 4-23
- multiplexed redo log files
 - total number of files, 5-2

N

- network configuration files
 - ldap.ora.ora, 8-12
 - listener.ora, 7-4, 8-10
 - Oracle Enterprise Manager service discovery, B-4
 - sqlnet.ora, 5-19, 8-16, B-6
 - testing, 9-12
 - tnsnames.ora, 5-19, 8-12
 - Oracle Enterprise Manager service discovery, B-5
- network directory, A-2, A-3
- New Database
 - configuration type, 3-6
- new features, xxvii
- nmiconf.log file, B-7
- Node Selection page, 4-7

O

- Operations page, 4-6, 4-20
- OPS_INSTANCE entry, B-7
- OPS_INSTANCE entry in services.ora file, B-7
- Optimal Flexible Architecture (OFA), 3-8, 4-2
- Oracle Enterprise Manager
 - architecture, 7-5
 - Configuration Assistant, 7-8
 - configuring, 7-10, 7-11
 - repository, 7-8
 - described, 7-2
 - GSD, 7-2
 - installing repository database, 7-8
 - nmiconf.log file, B-7
 - oratab file requirements, 5-13
 - repository database
 - configuring, 7-8
 - Server Management
 - Communication Daemon, 7-2
 - service discovery, B-2
 - services.ora file, B-6
 - setup scenarios, 7-5
 - sid_LIST_listener_name parameter in listener.ora file, 5-19, 7-4
 - solving service discovery failures, B-2
 - specifying preferred credentials, 7-8
 - SYSDBA privilege, 7-10
 - SYSOPER privilege, 7-10
 - TOOLS tablespace, 2-6
 - understanding discovery failures, B-2
- Oracle Managed Files
 - requirements for using, 1-6
- Oracle Performance Manager
 - accessing Real Application Clusters charts, 7-11
 - Cluster Database Instance folder, 7-11
 - configuring, 7-11
 - Databases folder, 7-12
- Oracle Universal Installer, 3-2
 - for Real Application Clusters, 3-1
 - overview of processing, 1-3
 - starting, 3-2
- ORACLE_SID
 - environment variable, 5-13
- OracleServicesid service, 5-14

- ORAPWD utility, 5-16
- orapwsid file, 5-16

P

- parameters
 - database creation, 5-2
 - initialization, 6-1
- password files, 5-16
- preconfigured database installation types, 3-7
- preferred credentials, 7-8
 - creating an operating system account, 7-9
 - database, 7-8
 - nodes, 7-8
- pre-installation
 - raw device creation, 2-13
- preinstallation, 2-12
 - on UNIX, 2-11
 - raw device creation, 2-3, 8-6
- primary and secondary instances
 - client configuration, 9-7
 - connecting to secondary instances, 9-8
 - initdb_name.ora file configuration, 9-7
 - listener configuration, 9-8
 - overview, 9-6
- private rollback segments, 5-6
 - creating, 5-6
- private threads, 5-8
- public rollback segments, 5-6
 - bringing online, 5-7
 - creating, 5-6
 - owner, 5-6
 - specifying, 5-7
 - using by default, 5-6
- PUBLIC threads, 5-8
- pwdsid.ora file, 5-16

R

- RAC_DATABASE entry, B-7
- RAC_DATABASE entry in services.ora file, B-7
- raw devices
 - configuration raw device, 2-2
 - configuring, 2-7
 - control files, 2-6

- creating
 - on UNIX, 2-7
 - on Windows, 2-12
- datafiles, 2-5
- DRSYS tablespace, 2-6
- environment variable, 2-10
- EXAMPLE tablespace, 2-6
- identifying on UNIX
 - control1 database object, 2-9
 - control2 database object, 2-9
 - drsys1 database object, 2-9
 - indx1 database object, 2-9
 - system1 database object, 2-9
 - temp1 database object, 2-9
 - tools1 database object, 2-9
 - users1 database object, 2-9
- identifying on Windows
 - db_name_control1 symbolic link, 2-13
 - db_name_control1 symbolic link, 8-7
 - db_name_control2 symbolic link, 2-13, 8-7
 - db_name_cwmlite1, 2-13
 - db_name_drsys1 symbolic link, 2-13, 8-6
 - db_name_indx1 symbolic link, 2-13, 8-5
 - db_name_rbs1 symbolic link, 8-5
 - db_name_redo_thread_number symbolic link, 2-13, 8-6
 - db_name_redothread_number symbolic link, 8-6
 - db_name_sfile1 symbolic link, 2-13, 8-5
 - db_name_system1 symbolic link, 2-13, 8-5
 - db_name_temp1 symbolic link, 2-13, 8-5
 - db_name_tools1 symbolic link, 2-13, 8-5
 - db_name_users1 symbolic link, 2-13, 8-5
- identifying on Windows, srvcfg, 2-13
- identifying on Windows, undotbs1 and undotbs2
 - symbolic links, 2-13, 8-5
- INDX tablespace, 2-6
- redo log files, 2-6
- server parameter files, 2-5
- setting up, 2-3, 2-13, 8-6
- srvcfg, 2-6
- SYSTEM tablespace, 2-5
- TEMP tablespace, 2-5
- TOOLS tablespace, 2-6
- UNDOTBS tablespace, 2-5
- USERS tablespace, 2-5
- verification, 4-3
- raw devices CWMLITE tablespace, 2-6
- raw volumes
 - on UNIX, configuring, 2-7
- RBS tablespace
 - description, 8-4
- Real Application Clusters
 - components, 1-2
 - databases, backing up, 5-18
 - databases, creating
 - with manual methods, 5-9
 - databases, deleting, 4-19
 - databases, password files, 5-16
 - hardware requirements, 1-6
 - installation requirements, 1-5
 - installed components of, 1-5
 - overview, 1-1, 4-1, 8-1
 - preinstallation, 2-12
 - raw device setup, 2-3, 2-13, 8-6
 - software requirements, 1-6
- Real Application Clusters Guard II, 9-9
- recovery
 - archive history, 5-2
 - instance, 5-2
 - starting another instance, 5-2
- redo log
 - changing, 5-9
- redo log files, 1-5, 2-2
 - described, 8-6
 - identifying on UNIX, 2-9
 - raw devices, 2-6
 - reconfiguring, 5-9
- redo log groups, creating, 5-8
- redo log mode
 - setting, 5-8
- registration
 - cross-node, 8-8
- registry
 - PM\db_name key, B-4
- repository database
 - installing, 7-8
- requirements
 - for installing Real Application Clusters, 1-5
 - tablespace sizes, 2-5

- restrictions
 - changing the redo log, 5-9
- .rhosts file, 2-11
- rollback segments, 5-5
 - described, 8-7
 - ID number, 5-6
 - manually creating, 5-5
 - multiple, 5-5
 - name, 5-6
 - public, 5-7
 - public versus private, 5-7
 - specifying, 5-6
 - tablespace, 5-5, 5-6
- ROLLBACK_SEGMENTS parameter
 - private and public segments, 5-6, 5-7
- root, 2-11

S

- Save as a Database Template
 - option on Creation options page, 4-17
- segments
 - ID number, 5-6
- server configuration
 - raw device for, 2-2
- Server Management
 - configuring, 7-1
 - GSD process, 7-2
 - node requirements, 7-4
 - requirements, 7-4
 - SIDs, 7-4
- server parameter file
 - backing up, 6-7
- server parameter files, 1-5, 2-2, 6-1, 6-2, 8-7
 - creating, 6-7
 - errors, 6-6
 - raw devices, 2-5
- service discovery
 - db_name.conf file, B-3
 - errors, B-2
 - listener.ora file, B-4
 - nmiconf.log file, B-7
 - on UNIX, B-3
 - on Windows NT, B-4
 - PM\db_name registry key, B-4
 - services.ora file, B-6
 - sid_LIST_listener_name parameter, B-4
 - tnsnames.ora file, B-5
 - understanding, B-2
- service registration
 - configuring, 8-8
- services.ora file, B-6, B-7
- shared configuration file, 8-2
- shared disks
 - configuring, 2-1
- shared server, 8-8
- shutting down a Server database
 - with SYSDBA privilege, 7-10
 - with SYSOPER privilege, 7-10
- sid Prefix, 3-9, 3-10
- sid_DESC parameter, B-4
- sid_LIST_listener_name parameter, 5-19, 7-4
- Software Only
 - configuration type, 3-6
- software requirements, 1-6
- spfile1
 - identifying on UNIX, 2-9
- sqlnet.ora file, 5-19, 8-16, B-6
 - default configuration, 8-16, B-6
- srvcfg
 - raw devices, 2-6
- srvcfg raw device
 - identifying, 2-13
- srvconfig
 - command on UNIX, 8-2
- srvConfig.loc, 8-2
- SRVCTL Utility, 5-13, 7-2
 - architecture on UNIX, 7-3
- srvm\admin directory, A-3
- SRVM_SHARED_CONFIG
 - environment variable, 3-2
- startup
 - during instance recovery, 5-2
 - rollback segments, 5-6
 - SYSDBA privilege, 7-10
 - SYSOPER privilege, 7-10
- Summary dialog, 4-18, 4-22
- SYSDBA privilege, 7-10
- SYSDBA privilege, granting, 7-9
- SYSOPER privilege, 7-10

SYSOPER privilege, granting, 7-9
SYSTEM tablespace, 5-5
 description, 8-4
 raw device, 2-5

T

tablespaces

and the DBCA, 8-4
control files, 2-6
CWMLITE, 2-6, 8-5
DRSYS, 2-6, 8-4
EXAMPLE, 2-6, 8-5
expanding for large sorts, 8-4
INDX, 2-6, 8-4
ODM, 2-6
RBS, 8-4
recommended sizes of, 2-5
rollback segment, 5-5, 5-6
server parameter file, 2-5
srvcfg, 2-6
SYSTEM, 2-5, 5-5, 8-4
TEMP, 2-5, 8-4
TOOLS, 2-6, 8-4
undo tablespaces for automatic undo
 management, 8-4
UNDOTBS, 2-5
USERS, 2-5, 8-4
XML, 2-6

TAF

configuring, 9-2
GLOBAL_DBNAME parameter in listener.ora
 file, 8-11
overview, 9-2
pre-establishing a connection, 9-4
retrying a connection, 9-4
 with client load balancing, 9-3
 with connect-time failover, 9-3

TEMP tablespace

described, 8-4
raw device, 2-5

testing the network, 9-12

THREAD clause, 5-8
 disabling a thread, 5-8
 when required, 5-8

THREAD parameter, 5-7

threads

associated with an instance, 5-7
changing from public to private, 5-8
creating, 5-7
disabled, 5-8
disabling, 5-8
public, 5-7, 5-8

tnsnames.ora file, 5-19, 8-12

default configuration, 8-12
FAILOVER parameter, 9-11
FAILOVER_MODE, 9-2
LOAD_BALANCE parameter, 9-11
service discovery, B-5

TOOLS tablespace

described, 8-4
raw device, 2-6

Transaction Processing

configuration type, 3-5
description of configuration type, 3-6

Transmission Control Protocol/Interconnect Protocol (TCP/IP)

for Real Application Clusters, 1-7
Transparent Application Failover, 9-2
 configuring, 9-2
 GLOBAL_DBNAME parameter in listener.ora
 file, 8-11
 overview, 9-2
 pre-establishing a connection, 9-4
 retrying a connection, 9-4
 with client load balancing, 9-3
 with connect-time failover, 9-3

troubleshooting

Database Configuration Assistant, 4-5
service discovery, B-2

U

undo management, 8-7

UNDOTBS tablespace

raw devices, 2-5

undotbs1

identifying on UNIX, 2-9

UNIX

raw volume names ASCII file, 2-10

- raw volumes, configuring, 2-7
- upgrading
 - Real Application Clusters issues, 4-23
- user
 - PUBLIC, 5-6
- User Datagram Protocol (UDP)
 - for Real Application Clusters, 1-7
- USERS tablespace
 - described, 8-4
 - raw device, 2-5

V

- VSSESSION table, 9-6
- Veritas Volume Manager
 - during installation, 2-5
- Virtual Interface Architecture (VIA)
 - for Real Application Clusters, 1-7
- volumes
 - Veritas Volume Manager, use of during installation, 2-5

W

- warming the cache, 9-9
- Welcome page, 4-4