Oracle9*i* Real Application Clusters

Installation and Configuration

Release 1 (9.0.1)

June 2001

Part No. A89868-01

Oracle9i Real Application Clusters Installation and Configuration, Release 1 (9.0.1)

Part No. A89868-01

Primary Author: Mark Bauer.

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

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

Graphic Designer: Valarie Moore.

Contents

Part I Installing Oracle9i Real Application Clusters

1 Introduction to Real Application Clusters Installation and Configuration

2 Configuring the Shared Disks

3 Using the Oracle Universal Installer for Real Application Clusters

4 Creating Databases with the Oracle Database Configuration Assistant

6 Configuring the Server Parameter File in Real Application Clusters Environments

Part II Oracle Enterprise Manager Installation Highlights

7 Installation and Configuration Highlights for Oracle Enterprise Manager in Real Application Clusters

Part III Managing the Configuration

8 Understanding The Installed Configuration

9 Configuring High Availability Features for Real Application Clusters

Part IV Migration

10 Migrating to Real Application Clusters

Part V Reference

A Directory Structure for Real Application Clusters Environments

B Oracle Enterprise Manager in Real Application Clusters Reference

Glossary

Index

List of Figures

List of Tables

Send Us Your Comments

Oracle9i Real Application Clusters Installation and Configuration, Release 1 (9.0.1)

Part No. A89868-01

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 Installation and Configuration explains how to install and configure Real Application Clusters environments. 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.

Note: Previous releases of Oracle9*i* Real Application Clusters are known as *Oracle Parallel Server.*

See Also:

- The *Oracle9i Real Application Clusters Documentation Online Roadmap* to navigate within the online Oracle9*i* 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

Oracle9i Real Application Clusters Installation 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*. That book also contains conceptual information about Oracle Real Application Clusters Guard.

Organization

This document contains the following five parts:

Part I: Installing Oracle9i Real Application Clusters

Part One 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.

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 Oracle Database Configuration Assistant"

This chapter explains how to use the Oracle Database Creation 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 the server parameter file for Real Application Clusters databases.

Part II: Oracle Enterprise Manager Installation Highlights

Part Two describes installation issues for installing Oracle Enterprise Manager in Real Application Clusters environments.

Chapter 7, "Installation and Configuration Highlights for Oracle Enterprise Manager in Real Application Clusters"

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

Part III: Managing the Configuration

Part Three 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"

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: Migration

Part Four provides information about migrating to Real Application Clusters.

Chapter 10, "Migrating to Real Application Clusters"

This chapter describes how to migrate to Real Application Clusters.

Part V: Reference

Part Five 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, "Oracle Enterprise Manager in Real Application Clusters Reference"

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*
- *Oracle Real Application Clusters Guard Administration and Reference Guide*

Installation Guides

- Your platform-specific Oracle Real Application Clusters Guard installation guide
- *Oracle9i Installation Guide* for Compaq Tru64, Hewlett-Packard HPUX, IBM-AIX, Linux, and Sun Solaris-based systems
- *Oracle9i Database installation guide for Windows*
- *Oracle Diagnostics Pack Installation*

Operating System-Specific Administrative Guides

- *Oracle9i Administrator's Reference* for Compaq Tru64, Hewlett-Packard HPUX, IBM-AIX, Linux, and Sun Solaris-based systems

- *Oracle9i Database Administrator's Guide for Windows*

Oracle9*i* 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*
- *Oracle 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://technet.oracle.com/membership/index.htm`

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://technet.oracle.com/docs/index.htm`

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 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**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.
`UPPERCASE monospace (fixed-width font)`	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a `NUMBER` column. 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.

Convention	Meaning	Example
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 data files 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.
lowercase monospace (fixed-width font) italic	Lowercase monospace italic font represents placeholders or variables.	You can specify the *parallel_clause*. Run U*old_release*.SQL where *old_release* refers to the release you installed prior to upgrading.

Conventions in Code Examples

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

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

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

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (*digits* [, *precision*])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE | DISABLE}
|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]

Convention	Meaning	Example
. . .	Horizontal ellipsis points indicate either:	
	■ That we have omitted parts of the code that are not directly related to the example	CREATE TABLE ... AS *subquery*;
	■ That you can repeat a portion of the code	SELECT *col1*, *col2*, ... , *coln* FROM employees;
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	

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;

Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*

UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees;
		SELECT * FROM USER_TABLES;
		DROP TABLE hr.employees;

lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		SELECT last_name, employee_id FROM employees;
		sqlplus hr/hr
		CREATE USER mjones IDENTIFIED BY ty3MU9;

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at:

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

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.

What's New in Installation and Configuration?

This section describes the new features of Oracle9*i* release 1 (9.0.1) as they pertain to the installation and configuration of **Oracle Real Application Clusters**.

See Also:

- *Oracle9i Real Application Clusters Concepts* for a detailed explanation of Real Application Clusters' and Oracle Real Application Clusters Guard's new features
- Your platform-specific documentation for information about installing Oracle Real Application Clusters Guard

The topic in this section is:

Oracle9*i* Release 1 (9.0.1) New Features for the Installation and Configuration of Real Application Clusters

The Oracle9*i* release 1 (9.0.1) Real Application Clusters features and enhancements described in this section are part of an effort to simplify the installation and configuration process. In addition, there is a feature to improve failover and a notice about the expiration of certain user names upon installation.

- **Oracle Cluster Setup Wizard for Windows operating systems**

  The Oracle Cluster Setup Wizard simplifies the installation of the Oracle **operating system-dependent clusterware (OSD)**.

  See Also:

  - Chapter 2, "Configuring the Shared Disks"
  - *Oracle9i Real Application Clusters Administration* for procedures on adding nodes and **instances**
  - *Oracle9i Database installation guide for Windows*

- **New raw device requirements**

  You must create a shared **raw device** to store Real Application Clusters server configuration information. In previous releases, Oracle stored this information in the `*db_name*.conf` file on UNIX and in the Registry on Windows NT platforms.

  If you use the preconfigured database templates, you must also create several shared raw devices that were optional in previous releases. Create these raw devices for the Oracle9*i* Sample Schemas, Online Analytical Processing (OLAP), Oracle9*i* *inter*Media, and for the **automatic undo management** feature. You must still create the other raw devices that were required in previous releases such as those for the `SYSTEM`, `USERS`, `TEMP`, and other tablespaces.

  See Also: Chapter 2, "Configuring the Shared Disks"

- **The Oracle Universal Installer and the Oracle Database Configuration Assistant have been enhanced to offer simplified software installation and database creation for Real Application Clusters**

  > **Note:** Oracle Corporation strongly recommends that you use the **Oracle Database Configuration Assistant (DBCA)** to create your Oracle Real Application Clusters database. Advanced features of Oracle9*i*, such as the **server parameter file** and automatic undo management, are best configured using the DBCA. If you use manual configuration procedures, make sure you have accurately completed the setup and configuration steps described in this manual and any other required procedures that are in other manuals as referred to in this document.

  The **Oracle Universal Installer (OUI)** offers the following database configuration types: **General Purpose**, **Transaction Processing**, **Data Warehouse**, **Customized**, and **Software Only**. The **Customized** database configuration type replaces the Custom installation type.

  The Oracle Database Configuration Assistant also has templates named General Purpose, Transaction Processing, and Data Warehouse. These templates correspond to the Installer's configuration types. In addition, the DBCA has a fully customizable template called **New Database** that does not include datafiles.

  To simplify installation, you can select the General Purpose, Transaction Processing, or Data Warehouse configuration type in the Universal Installer's Database Configuration screen. If you configured your shared raw devices as described in this book, after completing the remaining Installer screens the installation and DBCA database creation processes continue without further input.

  > **See Also:** Chapter 4, "Creating Databases with the Oracle Database Configuration Assistant"

- **The Oracle Database Configuration Assistant has Instance and Template Management features**

  Use Instance Management to add or delete an instance to or from a Real Application Clusters database. Use Template Management to manage database templates. You can also use Template Management to create templates from existing databases and then use these templates to create new databases.

- **The Oracle Database Configuration Assistant creates the server parameter file**

  Oracle Corporation recommends that you use the server parameter file when you install Real Application Clusters.

  > **See Also:** Chapter 4, "Creating Databases with the Oracle Database Configuration Assistant"

- **Modify the default file location for the server parameter file if using raw devices**

  Oracle Corporation recommends that you modify the default file location for the server parameter file for Real Application Clusters environments if you are using raw devices.

  > **See Also:** Chapter 4, "Creating Databases with the Oracle Database Configuration Assistant"

The following feature improves failover processing.

- **Using the DBMS_LIBCACHE Package in Primary/Secondary Instance Configurations**

To improve failover performance in Primary/Secondary instance configurations, use the DBMS_LIBCACHE package to transfer information from the library cache of the primary instance to the library cache of the secondary instance. This is known as **warming the library cache**.

> **See Also:** Chapter 9, "Configuring High Availability Features for Real Application Clusters"

The following notice applies to user names that expire upon installation.

- **Default user names expire as of this release**

Effective as of this release, all default user names *except* SYS, SYSTEM, and SCOTT, expire upon install. To use these names, you must explicitly unlock them.

Part I

Installing Oracle9*i* Real Application Clusters

Part One begins with an overview of the components required to install and configure Oracle9*i* Real Application Clusters and then provides procedures for:

- Configuring your disk subsystem
- Installing the Oracle9*i* Real Application Clusters software
- Creating your Real Application Clusters database
- Configuring the server parameter file

Part One also describes how to manually create your Real Application Clusters database. The chapters in Part One are:

1

Introduction to Real Application Clusters Installation and Configuration

This chapter provides an overview of the procedures and components involved in 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** software that facilitates inter-**node** communication. This section, with Figure 1–1, briefly describes the process for installing this software.

The Oracle9*i* Enterprise Edition and Oracle9*i* 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:

- The templates to create Real Application Clusters preconfigured databases
- The default **server parameter file** known as **spfile.ora** with its default settings for cluster databases
- Additional **redo log files** for operating cluster databases

The Operating System-Dependent Clusterware

The operating system clusterware contains **operating system-dependent clusterware (OSD)** components. The OSD clusterware controls the operating system and clusterware services required for Real Application Clusters. Its two primary subcomponents are the **Cluster Manager (CM)** and the **Inter-process Communication (IPC)** software.

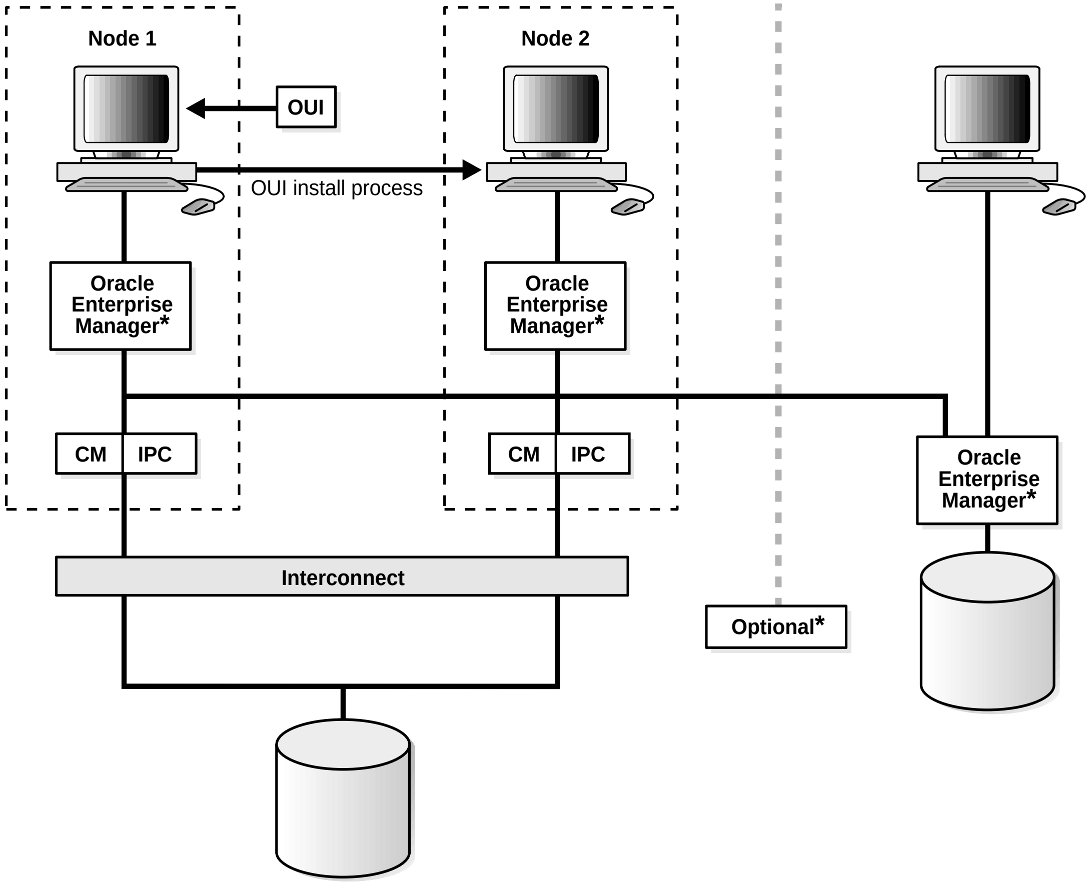
The Cluster Manager monitors the state of all nodes and broadcasts information about the nodes throughout the cluster. This includes information about which nodes are active or inactive members of the cluster. The IPC layer controls messaging functions so the nodes can communicate with each other through the interconnect.

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


Node 1
Node 2
OUI
OUI install process
Oracle Enterprise Manager*
Oracle Enterprise Manager*
Oracle Enterprise Manager*
CM
IPC
CM
IPC
Interconnect
Optional*

When the installer runs, Oracle Corporation recommends that you select one of the preconfigured databases or use the **Oracle Database Configuration Assistant (DBCA)** interactively to create your cluster database. You can also manually create your database with scripts provided you have properly configured your shared disks.

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

The Oracle Universal Installer can also install the software for the system management tool, **Oracle Enterprise Manager**, to integrate this tool into your new cluster environment. Or you can install Oracle Enterprise Manager onto other nodes outside your cluster (*). You can also use **Server Management (SRVM)** and **Performance Manager**, which are subcomponents of Oracle Enterprise Manager, to administer Real Application Clusters environments as described in the next section.

Server Management and Performance Manager for Real Application Clusters

You can use the Oracle Enterprise Manager Server Management (SRVM) subcomponent to start and stop Real Application Clusters databases. Use the Performance Manager subcomponent to monitor the performance of your Real Application Clusters database.

To operate, Oracle Enterprise Manager interacts with an **Oracle Intelligent Agent** which uses SRVM's **SRVCTL Utility** to control your Real Application Clusters database. Performance Manager interacts with the Intelligent Agent to monitor performance.

The Oracle Intelligent is installed when you install Oracle Enterprise Manager. The Oracle Intelligent Agent must reside on each node that is part of your Real Application Clusters database.

> **See Also:** *Oracle9i Real Application Clusters Administration* for information about using Server Management to administer Real Application Clusters environments and *Oracle9i Real Application Clusters Deployment and Performance* for information about using Performance Manager to monitor performance in Real Application Clusters environments

The Installed Real Application Clusters Storage Components

All instances in Real Application Clusters environments share the **control file**, **server parameter file**, and each **datafile**. These files reside on shared disks that are accessed by multiple nodes. Each instance has its own discrete set of **redo log files** that also reside on the shared disks. During failures, this arrangement of shared access to redo log files allows surviving instances to perform recovery for failed instances.

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. Configure the shared disks as described in Chapter 2.
3. Run the Oracle Universal Installer to install the Oracle9*i* Enterprise Edition and the Oracle9*i* Real Application Clusters software.
4. Create and configure your database as described in either:
   - Chapter 4 that explains how to use the Oracle Database Configuration Assistant (DBCA)
   - 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 Oracle9*i* 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 layer for UNIX, or Oracle operating system-dependent layers for Windows NT and Windows 2000
- ❏ Oracle9*i* Enterprise Edition
- ❏ Oracle Net
- ❏ Oracle Intelligent Agent release 1 (9.0.1) if using Oracle Enterprise Manager
- ❏ 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) If you use **Oracle Managed Files**, your platform must support a cluster file system and the cluster file system must be supported by Oracle. The Oracle Managed Files feature automatically creates and deletes files that Oracle requires to manage the database. Cluster file systems are available only for a limited number of system types.

> **See Also:** *Oracle9i Database Administrator’s Guide* for more information on 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 you use must be certified by Oracle for your Oracle9*i* 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 9.0.1 database with 64-bit instances. Depending on the platform, for example, 32-bit and 64-bit Oracle9*i* 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, proceed to Chapter 2 to configure the shared disks by either:

- Creating the raw volumes on UNIX, or
- Creating the logical drives on Windows NT and Windows 2000

2

Configuring the Shared Disks

This chapter describes how to configure shared disk subsystems to deploy **Real Application Clusters**. This chapter explains how to configure your subsystem so that the nodes within a **cluster** can read from and write to the shared disks. 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

> **Note:** Perform the procedures in this chapter only after ensuring that your system complies with the hardware and software requirements described in Chapter 1, and before continuing with the procedures for installing the **Oracle9i Enterprise Edition** software in Chapter 4.

Configuring Shared Disk Subsystems for Real Application Clusters

Real Application Clusters requires that all each **instance** be able to access a set of unformatted devices on a shared disk subsystem. These shared disks are also referred to as **raw devices**. If your platform supports an Oracle-certified cluster file system, however, you can store the files that Real Application Clusters requires directly on the cluster file system.

The Oracle instances in Real Application Clusters write data onto the raw devices to update the **control file**, **server parameter file**, each **datafile**, and each **redo log file.** All instances in the cluster share these files. 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 the 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 DBCA interactively, you must also configure additional raw devices as described under the heading "Additional Raw Devices Required by the DBCA" on page 2-3.

> **See Also:** The file size requirements as described in this chapter may change. Therefore, consult the README for the latest file size recommendations.

The Configuration Raw Device

You must create at least one shared raw device as an information repository for the database server configuration. You must do this:

- If this installation is the first Oracle9*i* Server installation on your cluster
- To migrate from a pre-release 1 (9.0.1) Oracle9*i* Real Application Clusters database to release 1 (9.0.1)
- To manually create your database

Oracle Enterprise Manager, the **Oracle Database Configuration Assistant (DBCA)**, the **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 previous 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 process for this raw device depends on your operating system:

- On UNIX, configure a raw device for the **Server Management (SRVM)** configuration raw device; the **Oracle Universal Installer (OUI)** prompts you for the name of this raw device.
- On Windows NT and Windows 2000, configure a **Cluster Manager (CM) voting disk** / SRVM Configuration Disk. Label this raw device `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 device as the Cluster Manager voting disk.

To use the DBCA to create your database, the procedures for which appear in Chapter 4, configure additional raw devices as described in the next section.

Additional Raw Devices Required by the DBCA

To use the Oracle 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 Oracle Universal Installer to install the Oracle Enterprise Edition software. The DBCA cannot create a Real Application Clusters database unless you have properly configured these raw devices:

- Seven for **datafiles**
- Two for control files
- One device for each instance for its own **tablespace** for **automatic undo management** (or one **rollback segment** tablespace for the database if you do not use automatic undo management)
- At least two for redo log files for each instance
- One for context
- Recommended: One for the server parameter file

- One for Oracle9*i inter*Media
- One for OLAP

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 log files, but all instances in a cluster share the control files and datafiles. Each instance's log files must be readable by all other instances for several types of recovery, such as instance 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 your plans for using each partition, determine the placement of these spare partitions using a couple of methods. For example, you can mix different sizes on one disk, or segment each disk into same-sized partitions.

Note: Ensuring that there are one or two spare partitions enables you to perform emergency file relocations or additions if a tablespace unexpectedly fills. For example, you can drop and re-create unused partitions into two or more partitions if needed.

DBCA Database Configuration Options

The DBCA supports several database templates for Real Application Clusters, **General Purpose, Transaction Processing**, **Data Warehouse**, and **New Database**. Chapter 3 describes these configuration types in more detail. The first three options are preconfigured database options and they include datafiles. As already mentioned, the New Database option does not include datafiles and is fully customizable.

If you use the General Purpose, Transaction Processing, or Data Warehouse configuration, you must create specific tablespaces using specific 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 file and block sizes. To use the Customized configuration option, ensure that the raw volumes on UNIX, or logical drives 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 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 both UNIX and Windows NT and Windows 2000 platforms. The preconfigured database options automatically create tablespaces and files with the capacities shown in Table 2–1. If you use the Customized database configuration, 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
`UNDOTBS` tablespace	580MB (total)
The DBCA uses automatic undo management by default. You should have one undo tablespace for each instance. **Note:** Divide the total undo space among the instances. For example, for a **General Purpose** database with two instances, create two undo tablespaces that are 290MB each. Regardless of how many instances you have, each undo tablespace should not be smaller than 50MB. If you do not use automatic undo management, then create an `RBS` tablespace for rollback segments and make this tablespace at least 580MB in size.	
`EXAMPLE` tablespace	160MB
`CWMLITE` tablespace	100MB
`INDX` tablespace	70MB
`TOOLS` tablespace	12MB
`DRSYS` tablespace	90MB
First control file	110MB
Second control file	110MB
Two redo log files for each instance	120MB (for each file)
`srvcfg` configuration raw device	100MB

Note: On Windows NT and Windows 2000, you can modify file sizes later, but raw device sizes are permanent after the DBCA creates datafiles on them. Therefore, an increase in file size does not expand the size of the raw device.

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, 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:** *Oracle8i 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 by using names that identify the database with the *db_name* entry, and that also relate to the raw volume type:

Table 2–2 Recommended Names for Oracle Database Files

Example File Name	Raw Volume
*db_name*_raw_system_400m	SYSTEM tablespace raw volume
*db_name*_raw_spfile_5m	Server parameter file raw volume
*db_name*_raw_users_120m	USERS tablespace raw volume
*db_name*_raw_temp_100m	TEMP tablespace raw volume
*db_name*_raw_undotbs1_290m	UNDOTBS 1 tablespace raw volume
*db_name*_raw_undotbs2_290m	UNDOTBS 2 tablespace raw volume
*db_name*_raw_example_30m	EXAMPLE tablespace raw volume
db_name_raw_cwmlite_100m	CWMLITE (OLAP) tablespace raw volume
*db_name*_raw_indx_70m	INDX tablespace raw volume
*db_name*_raw_tools_12m	TOOLS tablespace raw volume
*db_name*_raw_drsys_90m	DRSYS raw volume
*db_name*_raw_controlfile1_110m	First control file raw volume
*db_name*_raw_controlfile2_110m	Second control file raw volume
*db_name*_raw_*thread*_log*number*_120m	Two redo log files for each instance

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 to accommodate the RBS tablespace raw volume, substitute the entries for undotbs1 and undotbs2 with the following:

*db_name*_raw_rbs_580m

If you select the General Purpose, Transaction Processing, or Data Warehouse database configuration type on the Universal Installer's Database Configuration screen, or if you select the Customized option on this screen and then use one of the DBCA templates that uses datafiles, then you must perform the following step:

3. On the node from which you run the Oracle Universal Installer, create an ASCII file identifying the raw volume objects as shown in Table 2–3. The DBCA

requires that these objects exist during installation and 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 the raw volume object and *raw_device_file_path* is the path of the datafile, control file, or redo log file.

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

Database Object	**Used For**
system1	SYSTEM tablespace datafile
spfile1	Server parameter file
users1	USERS tablespace datafile
temp1	TEMP tablespace datafile
undotbs1	Undo tablespace 1 datafile
undotbs2	Undo tablespace 2 datafile
example1	EXAMPLE tablespace datafile
cwmlite1	CWMLITE (OLAP) tablespace datafile
indx1	INDX tablespace datafile
tools1	TOOLS tablespace datafile
drsys1	DRSYS tablespace datafile
control1	First control file
control2	Second control file
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

Use the following format for your undo tablespace files:

undotbs_*number*

Where *number* is the number of the instance.

Use the following format for your redo log files:

`redo`*`thread_number`*

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

Note: You must create at least two redo log file entries for each instance. If you do not use automatic undo management, you must make substitute entries for `undotbs1` and `undotbs2` with an entry for `rbs1` to use manual undo management.

When you create the ASCII file, separate the database objects from the paths with equals (=) signs as shown in Example 2–1:

Example 2–1 Example UNIX ASCII File for Preconfigured Database Configurations

```
system1=/dev/vx/rdsk/oracle_dg/clustdb_raw_system_400m
spfile1=/dev/vx/rdsk/oracle_dg/clustdb_raw_spfile_5m
users1=/dev/vx/rdsk/oracle_dg/clustdb_raw_users_120m
temp1=/dev/vx/rdsk/oracle_dg/clustdb_raw_temp_100m
undotbs1=/dev/vx/rdsk/oracle_dg/clustdb_raw_undotbs1_290m
undotbs2=/dev/vx/rdsk/oracle_dg/clustdb_raw_undotbs2_290m
example1=/dev/vx/rdsk/oracle_dg/clustdb_raw_example_30m
cwmlite1=/dev/vx/rdsk/oracle_dg/clustdb_raw_cwmlite_100m
indx1=/dev/vx/rdsk/oracle_dg/clustdb_raw_indx_70m
tools1=/dev/vx/rdsk/oracle_dg/clustdb_raw_tools_12m
drsys1=/dev/vx/rdsk/oracle_dg/clustdb_raw_dr_90m
control1=/dev/vx/rdsk/oracle_dg/clustdb_raw_controlfile1_110m
control2=/dev/vx/rdsk/oracle_dg/clustdb_raw_controlfile2_110m
redo1_1=/dev/vx/rdsk/oracle_dg/clustdb_raw_log11_120m
redo1_2=/dev/vx/rdsk/oracle_dg/clustdb_raw_log12_120m
redo2_1=/dev/vx/rdsk/oracle_dg/clustdb_raw_log21_120m
redo2_2=/dev/vx/rdsk/oracle_dg/clustdb_raw_log22_120m
```

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

```
rbs1=/dev/vx/rdsk/oracle_dg/clustdb_raw_rbs1_580m
```

Also make sure to use the following format for your redo log files:

```
redothread_number
```

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

> **Note:** You must create at least two redo log file entries and one undo tablespace for each instance.

You must specify that Oracle should use this file to determine the raw device partition locations by setting the following environment variable where *filename* is the name 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` account on each node so that the account:
   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 Oracle Universal Installer, 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` account user, check for user equivalence for the `oracle` account by performing a remote login (`rlogin`) to each node in the cluster.

7. As `oracle` account user, if you are prompted for a password, you have not given the *oracle* account the same attributes on all nodes. You must correct this because the Oracle Universal Installer cannot use the `rcp` 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 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 Oracle9*i* Enterprise Edition software and to configure your Oracle9*i* Real Application Clusters database.

Configuring Logical Drives on Windows NT and Windows 2000

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 disk. To configure the logical drives, create multiple logical partitions using Windows NT Disk Administrator or Windows 2000 Computer Manager.

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:

Table 2–4 Symbolic Link Names for Preconfigured Database Types

Symbolic Link Name...	Used for...
*db_name*_system1	SYSTEM tablespace datafile
*db_name*_spfile1	Server parameter file
*db_name*_users1	USERS tablespace datafile
*db_name*_temp1	TEMP tablespace datafile
*db_name*_undotbs1	UNDO tablespace 1
*db_name*_undotbs2	UNDO tablespace 2
*db_name*_example1	EXAMPLE tablespace
*db_name*_cwmlite1	CWMLITE tablespace
*db_name*_indx1	INDX tablespace datafile
*db_name*_tools1	TOOLS tablespace datafile
*db_name*_drsys1	DRSYS tablespace datafile
*db_name*_control1	First control file
*db_name*_control2	Second control file
srvcfg	SRVM Configuration Repository
*db_name*_redo_*thread_number* Where *thread* is the thread ID of the instance and *number* is the log number (1 or 2) for the instance.	redo log files Each instance must have two redo log files. If the database name is db, the link names for the first instance should be: db_redo1_1 db_redo1_2 The link names for the second instance should be: db_redo2_1 db_redo2_2

Note: If you do not use automatic undo management, then to accommodate a rollback segment tablespace, replace the `*db_name*_undotbs1` and `*db_name*_undotbs2` symbolic link names with:

*db_name*_rbs1

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 Oracle9*i* Enterprise Edition software and to configure your Real Application Clusters database.

3

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 are:

- Starting the Oracle Universal Installer
- Using the General Purpose, Transaction Processing, and Data Warehouse Database Configuration Types
- Using the Customized Database Configuration Type

Note: Perform the procedures in this chapter only after configuring your disk subsystem as described in Chapter 2, and before continuing with the procedures for using the **Oracle Database Configuration Assistant (DBCA)** as described in Chapter 4.

Starting the Oracle Universal Installer

Follow these procedures to use the Oracle Universal Installer to install the Oracle Enterprise Edition and the Real Application Clusters software.

1. Before running the Oracle Universal Installer, from the **node** where you intend to run the Installer, verify that you have administrative privileges on the other nodes using one of the operating system-specific verification methods in Table 3–1.

Table 3–1 Privilege Verification Methods

Operating System	Verification Method
On UNIX	Perform a remote copy (`rcp`) to the other nodes, including the local node.
On Windows NT and Windows 2000	Enter the following command for each node that is a part of the **cluster** where *`node_name`* is the node name returned from the `lsnodes.exe` utility that is in the `preinstall_rac` directory on the installation CD: `NET USE \\node_name\C$` 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.

Then for UNIX platforms only, set the environment variables as described in the following two steps, otherwise, 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 the configuration storage by entering, for example:

   ```
   setenv SRVM_SHARED_CONFIG shared_filename
   ```

   If your platform supports a cluster file system, 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 Raw Device Name page.

> **Note:** All installations must be identical across 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.
5. 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.
6. On the Available Products page, select **Oracle9*i***.
7. On the Install Types page, select **Enterprise Edition**. The Installer displays the Database Configuration page.

Before continuing, read the following description about how the Oracle Database Configuration Assistant creates your database and about the underlying configuration that it creates. This section also describes the configuration types.

However, if you are familiar with the DBCA and have decided on a database configuration type, proceed to one of the following sections:

- "Using the General Purpose, Transaction Processing, and Data Warehouse Database Configuration Types" on page 3-6
- "Using the Customized Database Configuration Type" on page 3-8

Selecting a Database Configuration Type

When you run the Oracle Universal Installer and select the Oracle9*i* 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.

Descriptions for Using the Configuration Types in Real Application Clusters

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 Oracle9*i* Real Application Clusters), networking services, Oracle9*i* utilities, and online documentation. At the end of the installation, the Database Configuration Assistant 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	Allows you 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.	—

See Also: The Oracle Universal Installer documentation for more detailed configuration type descriptions

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, complete the procedures in this chapter under the heading "Using the General Purpose, Transaction Processing, and Data Warehouse Database Configuration Types". 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.

The DBCA processing for these configuration types creates a starter database and configures the Oracle network services. As mentioned, if your platform does not support 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, however, you must enter specific information as described in the next section.

Using The Customized Configuration Type

If you select the Customized configuration type, the Oracle Universal Installer runs the Oracle Database Configuration Assistant 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 particular environment. You also can customize these templates as required.

The New Database type, however, creates a database without using preconfigured options. Therefore, 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 Oracle Database Configuration Assistant".

The following section provides more detail on Oracle Database Configuration Assistant processing when it creates a Real Application Clusters database.

Oracle Database Configuration Assistant Processing

The **Oracle Database Configuration Assistant (DBCA)** runs as part of the Oracle Universal Installer installation process. When the Installer begins processing, the **Oracle Net Configuration Assistant** runs. If you configured directory access with the Oracle Net Configuration Assistant, 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 OracleNet Configuration Assistant completes its processing, 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:

- 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

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

See Also:

- The *Oracle 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

Using the General Purpose, Transaction Processing, and Data Warehouse Database Configuration Types

Continuing from Step 7 on page 3-3, execute the following procedures to complete a General Purpose configuration:

1. On the Database Configuration page, select **General Purpose**, **Transaction Processing,** or **Data Warehouse**. If the Installer displays the Node Selection page, skip the next step and proceed to step 3.
2. If the Installer does not display the Node Selection page, 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/OraInstall` directory. Refer to your vendor's clusterware documentation if the output indicates that your clusterware is not properly installed.

- Windows NT and Windows 2000 Clusterware Diagnostics

  For Windows NT and Windows 2000 platforms, execute the `lsnodes -v` command from the `preinstall_rac` directory.

  ```
  lsnodes -v
  ```

  Refer to your clusterware documentation if the detailed output indicates that your clusterware is not running.

3. Select the nodes that you want 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**.
4. The Installer displays the Upgrading or Migrating an Existing Database page. Do not use this screen; it is not intended for Real Application Clusters. Click **Next**.
5. 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 that is to be used 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 `db1db1`, and the second instance is given a *`sid`* of `db2`. Click **Next**.
6. 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 complete the process.

During installation, the installer does not display messages indicating which components are installed on the remote nodes.

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 7 on on page 3-3, execute the following procedures to complete a Customized configuration:

1. On the Database Configuration page, select **Customized**. The Oracle Universal Installer displays the Node Selection page.
2. If the DBCA displays the Node Selection page, skip this step and proceed to step 3. Otherwise, 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/OraInstall` directory. Refer to your vendor's clusterware documentation if the output indicates that your clusterware is not properly installed.
   - Windows NT and Windows 2000 Clusterware Diagnostics

     For Windows NT and Windows 2000 platforms, execute the `lsnodes -v` command from the `preinstall_rac` directory.

     ```
     lsnodes -v
     ```

   Refer to your clusterware documentation if the detailed output indicates that your clusterware is not running.
3. Select the nodes that you want 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**.
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 Raw Device Name page. Enter the name of a raw device that is at least 100MB in size, or a cluster file system file on the Raw Device Name page and Click **Next**.
5. If the Installer detects a pre-existing Oracle database, the Installer displays the Upgrading or Migrating an Existing Database page. Do not use this page because it is not intended for Real Application Clusters. Click **Next**.

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 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 complete the process.

During installation, the Installer does not display messages indicating that the components are being installed on the other nodes.

When you complete these procedures, proceed to Chapter 4, "Creating Databases with the Oracle Database Configuration Assistant".

4

Creating Databases with the Oracle Database Configuration Assistant

This chapter describes using the **Oracle Database Configuration Assistant (DBCA)** to create and delete **Real Application Clusters** databases. It explains how to decide whether to manually create your database and how to use the DBCA to delete Real Application Clusters databases. This chapter also briefly discusses migration and multiple Oracle homes issues.

The topics in this chapter include:

- Using the Database Configuration Assistant
- Benefits of Using the Oracle 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 1 (9.0.1)
- 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:

- 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

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

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 *Oracle 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 Oracle 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 Oracle9*i* 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. So even if you have **datafile** requirements that differ from those offered in one of the DBCA templates, use the DBCA. You can also execute user-specified scripts as part of the database creation process.

The DBCA and the **Oracle Net Configuration Assistant** also accurately 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 do not create a database during installation, you can create one later using the DBCA in standalone mode. To do this, 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, 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, the DBCA cannot create your database.

To start the DBCA, on one of the nodes:

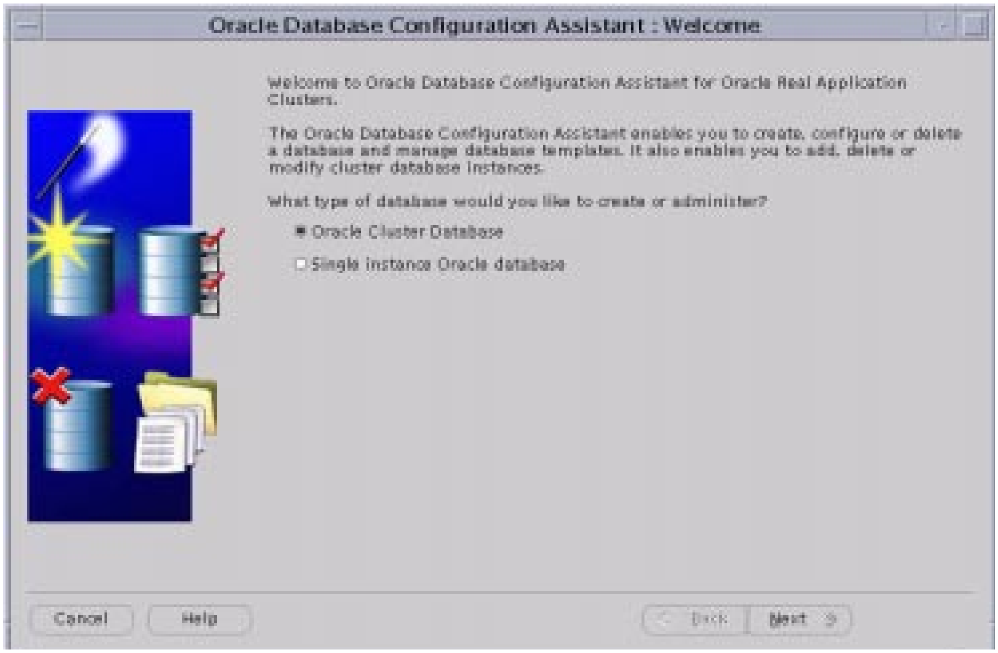
- On UNIX, enter the command `dbca` from the `$ORACLE_HOME/bin` directory
- On Windows NT and Windows 2000, choose Start > Programs > Oracle - [*HOME_NAME*] > Database Administration> Database Configuration Assistant

Creating a Real Application Clusters Database with the DBCA

The following section describes how to use the Database Configuration Assistant 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 as 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



Oracle Database Configuration Assistant : Welcome

Welcome to Oracle Database Configuration Assistant for Oracle Real Application Clusters.

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

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

● Oracle Cluster Database

○ Single instance Oracle database

> **See Also:** The Oracle Database Configuration Assistant online help for DBCA more information

If the DBCA does not display the Real Application Clusters Welcome page with the Oracle Cluster Database option, 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 `$ORACLE_HOME/bin` directory. Refer to your vendor's clusterware documentation if the output indicates that your clusterware is not properly installed. Resolve the problem and then restart the DBCA.

Windows NT and Windows 2000 Clusterware Diagnostics

For Windows NT and Windows 2000 platforms, execute the `lsnodes -v` command from the `%ORACLE_HOME%\bin` directory.

```
lsnodes -v
```

Refer to your clusterware documentation if the detailed output indicates that your clusterware is not running. Resolve the problem and then restart the DBCA.

> **Note:** If the DBCA Welcome page for Real Application Clusters does not display, it means the Database Configuration Assistant was unable to:
>
> - Detect whether the operating system specific clusterware is running on UNIX platforms
> - Load the Cluster Manager software on Windows NT and Windows 2000 platforms

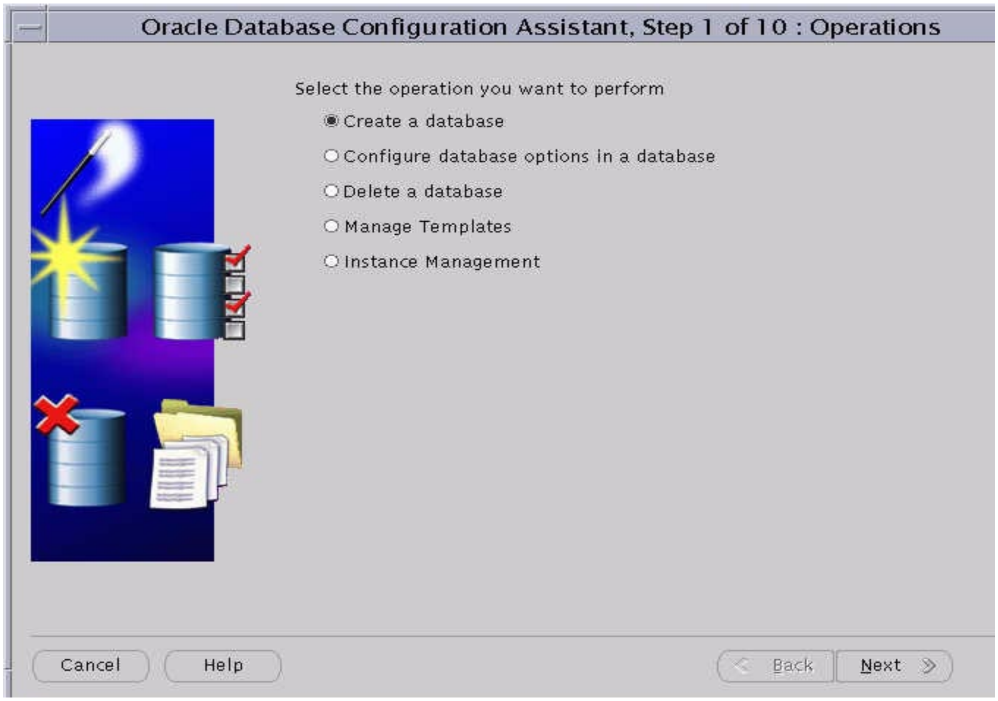
To create a Real Application Clusters database:

1. Select the **Oracle Cluster Database** option and click **Next**.

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

After you click **Next**, the DBCA displays the Operations page shown in Figure 4–2.

Figure 4–2 Operations



Oracle Database Configuration Assistant, Step 1 of 10 : Operations

Select the operation you want to perform

● Create a database

○ Configure database options in a database

○ Delete a database

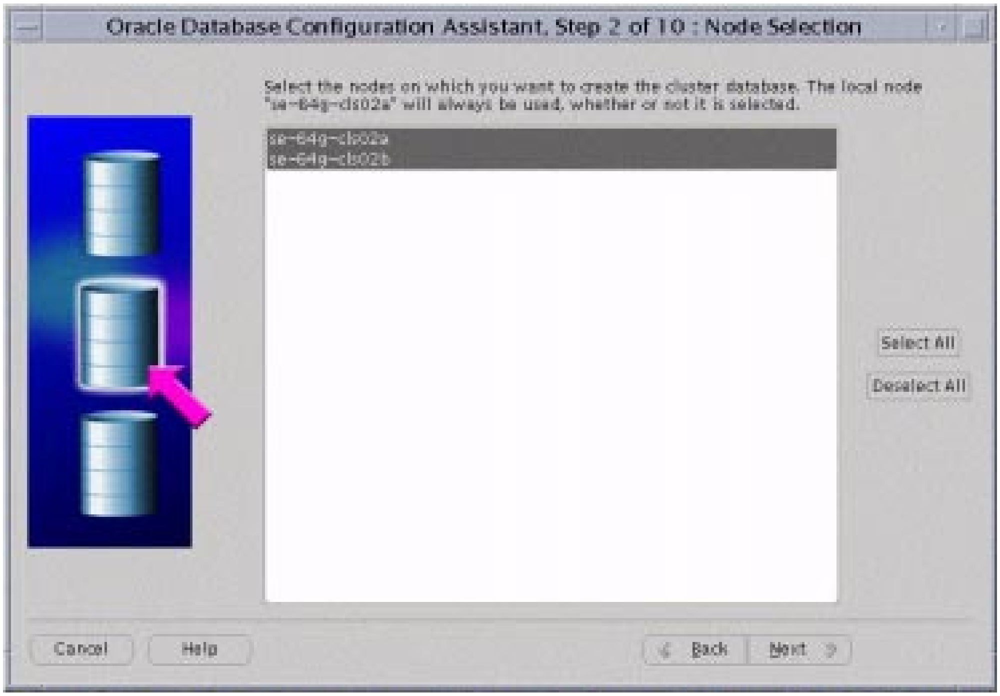
○ Manage Templates

○ Instance Management

2. Select the **Create a Database** option and click **Next**.

After you click **Next**, The DBCA displays the Node Selection page shown in Figure 4–3.

Figure 4–3 Node Selection Display Page



Oracle Database Configuration Assistant, Step 2 of 10 : Node Selection

Select the nodes on which you want to create the cluster database. The local node "se-64g-cls02a" will always be used, whether or not it is selected.

se-64g-cls02a
se-64g-cls02b

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

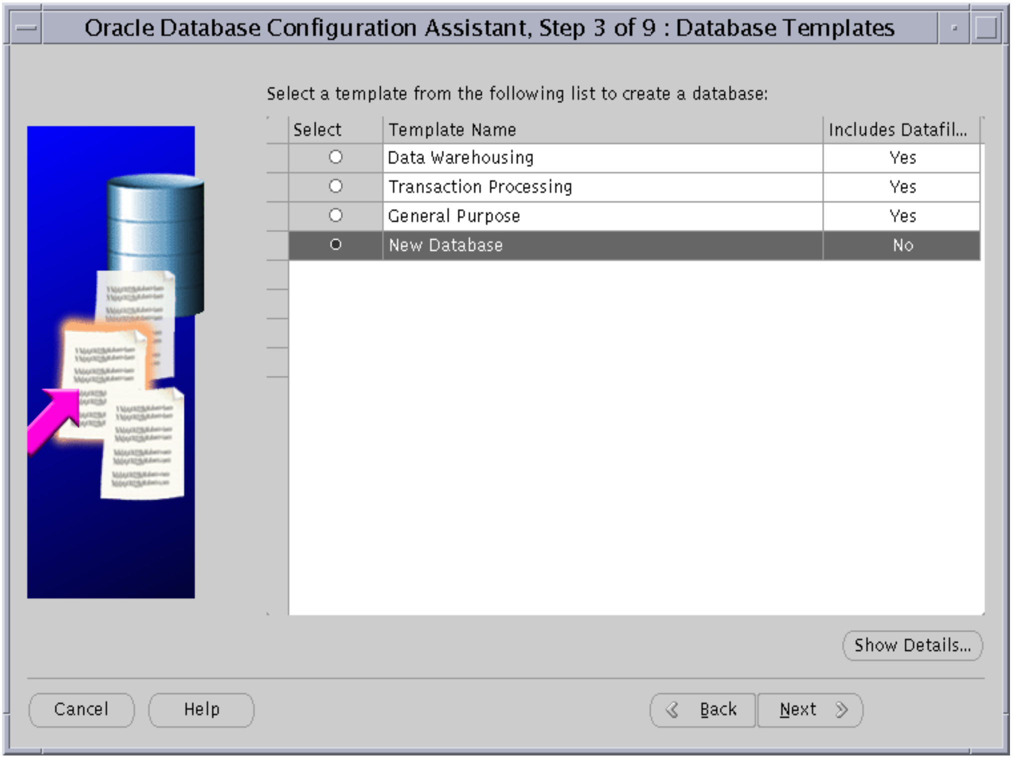
3. If nodes are missing from the Node Selection Display page, perform clusterware diagnostics as described under one of the following platform-specific headings, "UNIX Clusterware Diagnostics" on page 4-4, or "Windows NT and Windows 2000 Clusterware Diagnostics" on page 4-5.

4. Select the nodes that you want to configure as members of your cluster database and click **Next**. The local node is always selected by default.

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

After you click **Next**, the DBCA displays the Database Template Selection page as shown in Figure 4–4.

Figure 4–4 Template Selection Page



Oracle Database Configuration Assistant, Step 3 of 9 : Database Templates

Select a template from the following list to create a database:

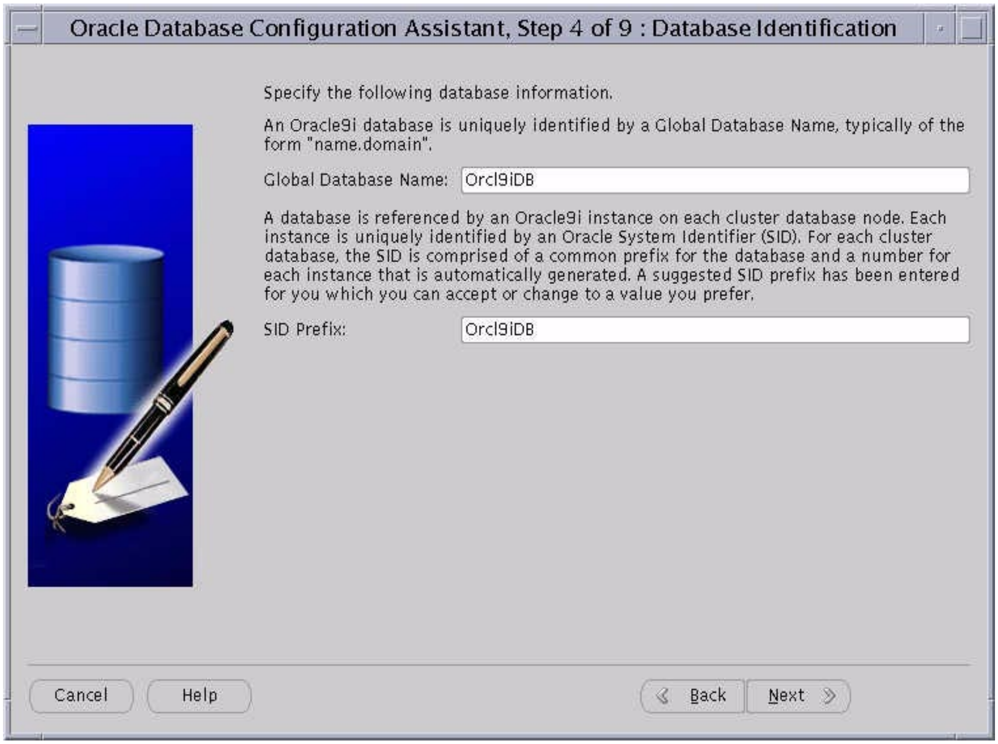
Select	Template Name	Includes Datafil...
○	Data Warehousing	Yes
○	Transaction Processing	Yes
○	General Purpose	Yes
○	New Database	No

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.

Select one of the four database templates shown on this page. These include the **Data Warehouse**, **Transaction Processing**, and **General Purpose** templates. These templates include datafiles. However, the fourth template, **New Database,** does not include datafiles.

After you click **Next**, the DBCA displays the Database Identification page as shown in Figure 4–5.

Figure 4–5 Database Identification Page



Oracle 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: Orcl9iDB

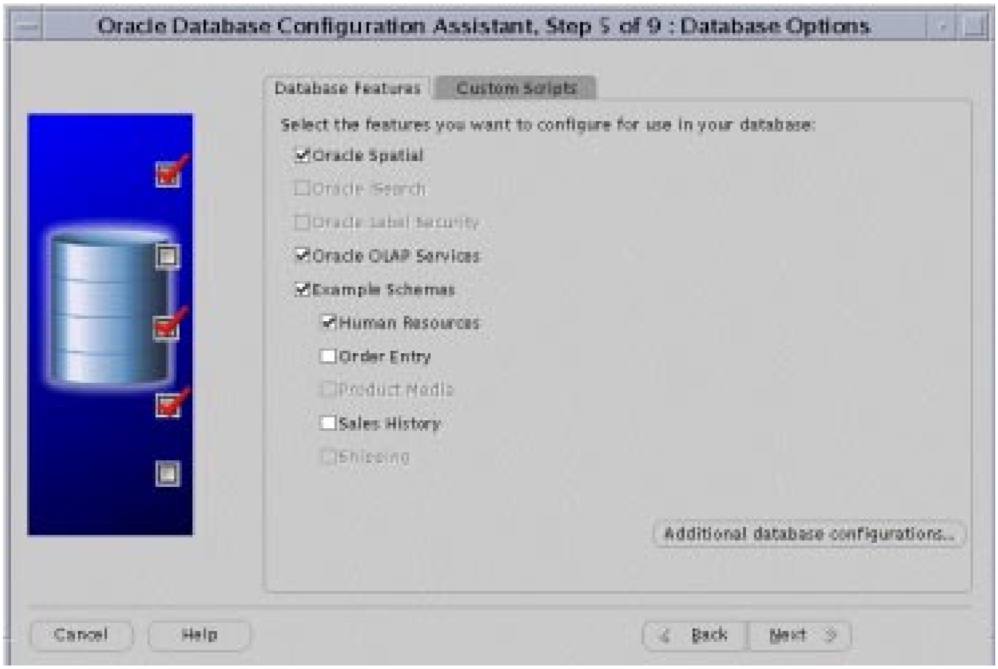
A database is referenced by an Oracle9i instance on each cluster database node. Each instance is uniquely identified by an Oracle System Identifier (SID). For each cluster database, the SID is comprised of a common prefix for the database and a number for each instance that is automatically generated.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

SID Prefix: Orcl9iDB

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

After you click **Next**, if you have selected the **New Database** template the DBCA displays the Database Options page as shown in Figure 4–5. If you selected one of the other preconfigured database options, then after you click **Next** the DBCA displays the Initialization Parameters page as shown in Figure 4–8 on page 4-12.

Figure 4–6 Database Options Page


Oracle Database Configuration Assistant, Step 5 of 9 : Database Options
Database Features
Custom Scripts
Additional database configurations...
Cancel
Help
Back
Next

Select the features you want to configure for use in your database:

☑ Oracle Spatial

□ Oracle Search

□ Oracle Label Security

☑ Oracle OLAP Services

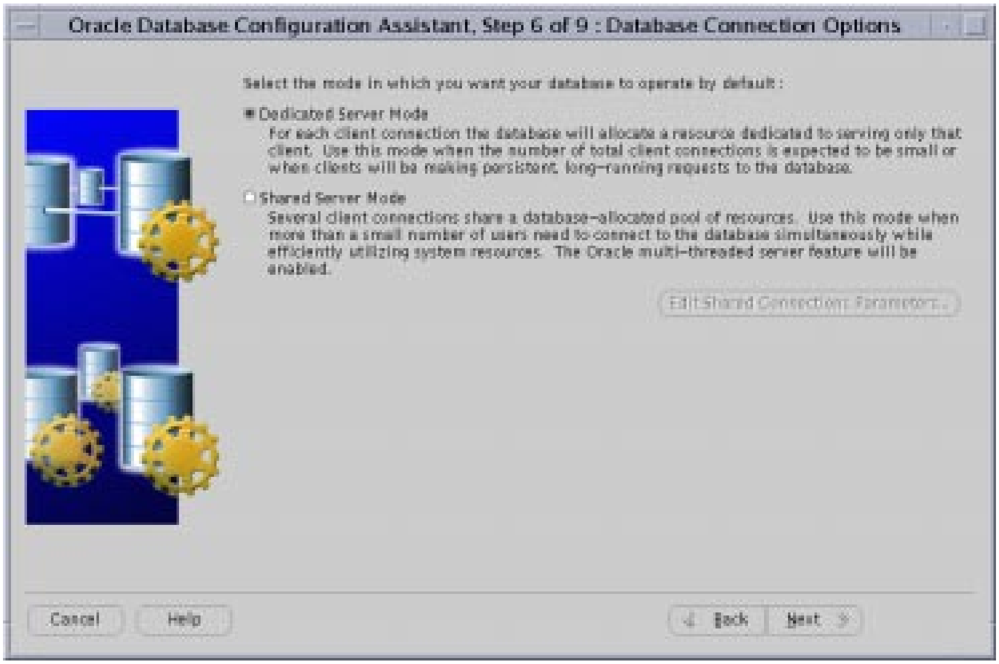
☑ Example Schemas

- ☑ Human Resources
- □ Order Entry
- □ Product Media
- □ Sales History
- □ Shipping

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

After you click **Next**, if you have selected the **New Database** template the DBCA displays the Database Connections page as shown in Figure 4–5.

Figure 4–7 Database Connections Options Page



Oracle Database Configuration Assistant, Step 6 of 9 : Database Connection Options

Select the mode in which you want your database to operate by default :

● Dedicated Server Mode

For each client connection the database will allocate a resource dedicated to serving only that client. Use this mode when the number of total client connections is expected to be small or when clients will be making persistent, long-running requests to the database.

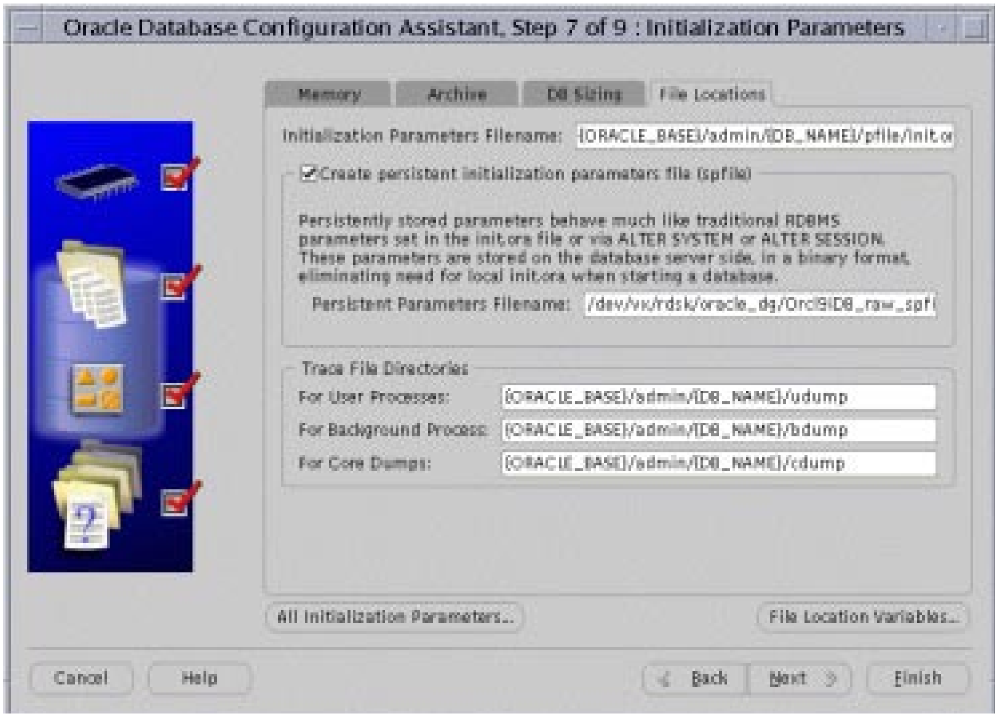
○ Shared Server Mode

Several client connections share a database-allocated pool of resources. Use this mode when more than a small number of users need to connect to the database simultaneously while efficiently utilizing system resources. The Oracle multi-threaded server feature will be enabled.

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

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

Figure 4–8 Initialization Parameters



Oracle Database Configuration Assistant, Step 7 of 9 : Initialization Parameters
Memory
Archive
DB Sizing
File Locations
All Initialization Parameters...
File Location Variables...
Cancel
Help
Back
Next
Finish

Initialization Parameters Filename: {ORACLE_BASE}/admin/{DB_NAME}/pfile/init.or

☑ Create persistent initialization parameters file (spfile)

Persistently stored parameters behave much like traditional RDBMS parameters set in the init.ora file or via ALTER SYSTEM or ALTER SESSION. These parameters are stored on the database server side, in a binary format, eliminating need for local init.ora when starting a database.

Persistent Parameters Filename: /dev/vx/rdsk/oracle_dg/Orcl9iDB_raw_spf

Trace File Directories

For User Processes: {ORACLE_BASE}/admin/{DB_NAME}/udump

For Background Process: {ORACLE_BASE}/admin/{DB_NAME}/bdump

For Core Dumps: {ORACLE_BASE}/admin/{DB_NAME}/cdump

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

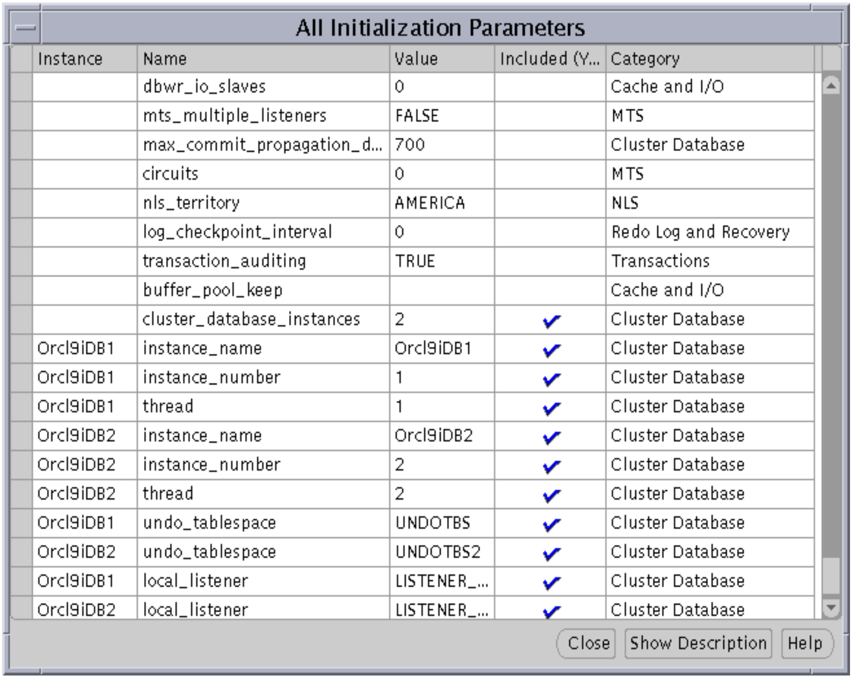
If you select the **Create persistent initialization parameter file (spfile)** option, which is equivalent to the server parameter file, you may need to modify the location for the server parameter file depending on the type of file system you use:

- If you use a cluster file system, or if you have a single-node Real Application Clusters environment, you can place the server parameter file on the file system.
- If you do not have 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 **Persistent Parameters Filename** field in the center of the Initialization Parameters page as shown in Figure 4–8.

10. Review the parameter settings that the DBCA will configure in the server parameter file by clicking the **All Initialization Parameters** option.

When you click **All Initialization Parameters**, the DBCA displays the All Initialization Parameters dialog in Figure 4–9.

Figure 4–9 All Initialization Parameters Dialog



Instance	Name	Value	Included (Y...	Category
	dbwr_io_slaves	0		Cache and I/O
	mts_multiple_listeners	FALSE		MTS
	max_commit_propagation_d...	700		Cluster Database
	circuits	0		MTS
	nls_territory	AMERICA		NLS
	log_checkpoint_interval	0		Redo Log and Recovery
	transaction_auditing	TRUE		Transactions
	buffer_pool_keep			Cache and I/O
	cluster_database_instances	2	✓	Cluster Database
Orcl9iDB1	instance_name	Orcl9iDB1	✓	Cluster Database
Orcl9iDB1	instance_number	1	✓	Cluster Database
Orcl9iDB1	thread	1	✓	Cluster Database
Orcl9iDB2	instance_name	Orcl9iDB2	✓	Cluster Database
Orcl9iDB2	instance_number	2	✓	Cluster Database
Orcl9iDB2	thread	2	✓	Cluster Database
Orcl9iDB1	undo_tablespace	UNDOTBS	✓	Cluster Database
Orcl9iDB2	undo_tablespace	UNDOTBS2	✓	Cluster Database
Orcl9iDB1	local_listener	LISTENER_...	✓	Cluster Database
Orcl9iDB2	local_listener	LISTENER_...	✓	Cluster Database

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

11. To review the instance-specific parameter settings, scroll down 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 Oracle should place the parameter setting in your **server parameter file**.

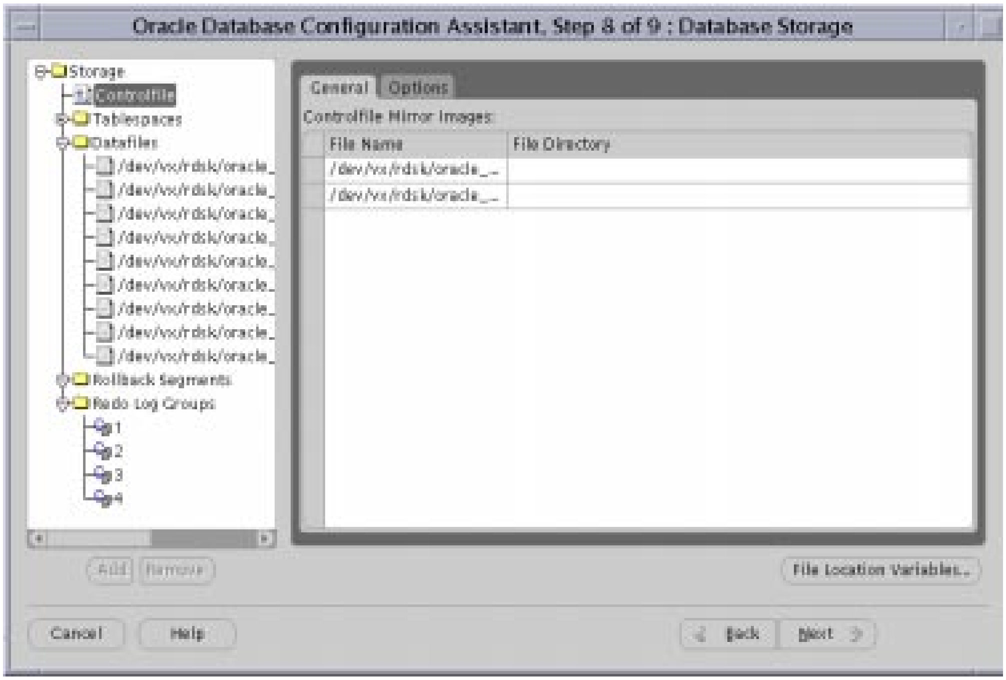
Oracle only places parameter entries from the All Initialization Parameters Dialog into the server parameter file that have check marks in the **Included (Y/N)** column. Also note the following points about the All Initialization Dialog:

- 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 page and click **Close**. Make sure your entries for the **File Locations** tab, as well as for 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 in Figure 4–10.

Figure 4–10 Database Storage Page



Oracle Database Configuration Assistant, Step 8 of 9 : Database Storage

Controlfile Mirror Images:

File Name	File Directory
/dev/vx/rdsk/oracle_...	
/dev/vx/rdsk/oracle_...	

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 file names for the **control files**, **datafiles**, and redo log groups on this page.
- For Windows NT and Windows 2000, the filenames are default symbolic link names. You can change these if you named your symbolic links differently from the Oracle recommendations. If you override these default symbolic link names, then enter names that incorporate your database name to simplify administration.

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

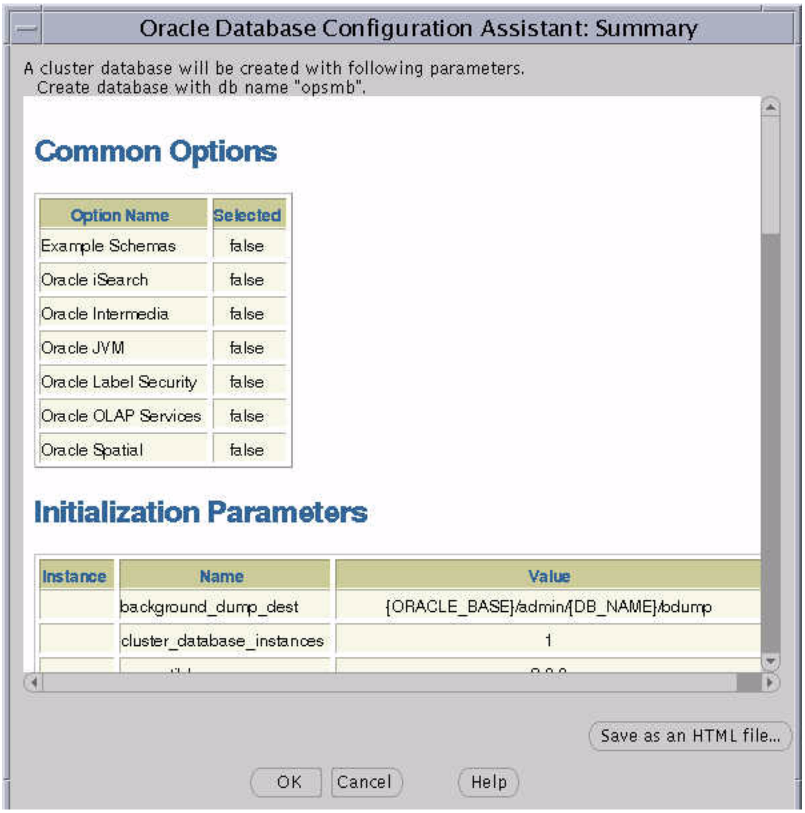
15. To rename temporary file names if you used a template that includes datafiles, click on the temporary file name and overwrite it.
16. Click **Next** when you have completed entering data on the Database Storage page.

After you click **Next**, the DBCA displays the Creation Options page on which you can select the database creation options you want to use. For example:

- **Create Database**—creates the database now
- **Save as a Database Template**—creates a template that captures 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 page similar to the page in Figure 4–11.

Figure 4–11 DBCA Summary Page



A cluster database will be created with following parameters.
Create database with db name "opsmb".

Common Options

Option Name	Selected
Example Schemas	false
Oracle iSearch	false
Oracle Intermedia	false
Oracle JVM	false
Oracle Label Security	false
Oracle OLAP Services	false
Oracle Spatial	false

Initialization Parameters

Instance	Name	Value
	background_dump_dest	{ORACLE_BASE}/admin/{DB_NAME}/bdump
	cluster_database_instances	1

17. Review the information on the Summary page and click **OK**. The DBCA then displays database creation progress indicators.

At this point in the installation 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 instances and listeners

Deleting a Database with the Database Configuration Assistant

This section explains how to use the Database Configuration Assistant (DBCA) to delete a database.

Using the DBCA to delete a database removes a database’s initialization parameter files, instances, OFA structure, and network configuration. However, the DBCA does not remove datafiles if you placed the files on raw partitions.

To delete a database with the DBCA:

1. Start the DBCA on one of the nodes:

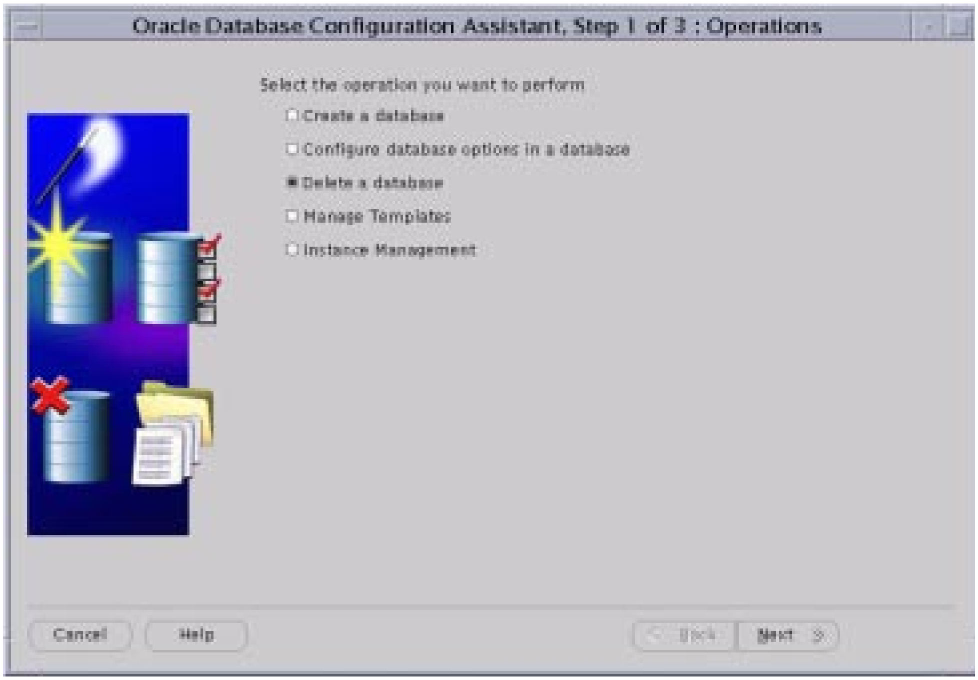
   - On UNIX, execute the `dbca` command from the `$ORACLE_HOME/bin` directory
   - On Windows NT and Windows 2000, choose Start > Programs > Oracle - [*HOME_NAME*] > Oracle Database Configuration Assistant.

   The Welcome page appears as shown earlier in this chapter in Figure 4–1.

2. Select the **Oracle Cluster Database** option and click **Next**.

After you click **Next**, The DBCA displays the Operations page as shown in Figure 4–12.

Figure 4–12 Operations



Oracle Database Configuration Assistant, Step 1 of 3 : Operations

Select the operation you want to perform

○ Create a database

○ Configure database options in a database

● Delete a database

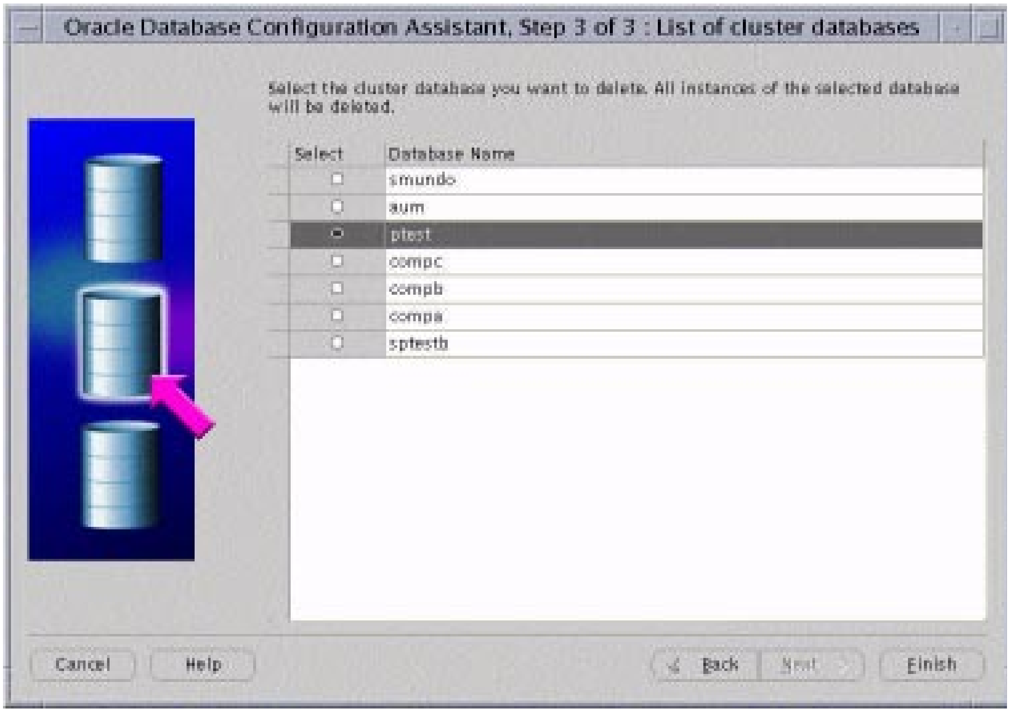
○ Manage Templates

○ Instance Management

3. Click the **Delete a database** option and click **Next**.

After you click **Next**, the DBCA displays the List of Databases page similar to the one in Figure 4–13.

Figure 4–13 Oracle Database Configuration Assistant Database List of Databases Page



Oracle Database Configuration Assistant, Step 3 of 3 : List of cluster databases

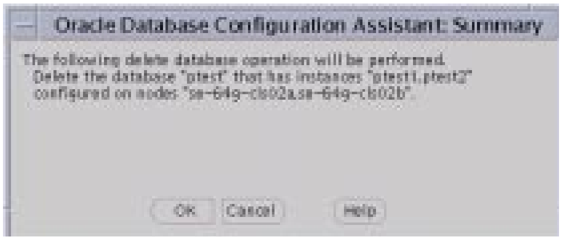
Select the cluster database you want to delete. All instances of the selected database will be deleted.

Select	Database Name
○	smundo
○	aum
●	ptest
○	compc
○	compb
○	compa
○	sptestb

4. If your user ID and password are not operating-system authenticated, then the List of 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 you want to delete and click **Finish**.

After you click **Finish**, the DBCA displays an alert dialog showing the database name and associated instances that the DBCA is going to delete. This alert dialog also displays the node or nodes on which the instances exist as shown in Figure 4–14:

Figure 4–14 Oracle Database Configuration Assistant Database Deletion Alert Dialog



The following delete database operation will be performed.
Delete the database "ptest" that has instances "ptest1,ptest2"
configured on nodes "sa-64g-cb02a,sa-64g-cb02b".

6. Click **OK** to complete 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, OracleService*sid* services, and oratab entries.

Migrating or Upgrading to Release 1 (9.0.1)

If the **Oracle Universal Installer (OUI)** detects an earlier version of Oracle, it prompts you to migrate or upgrade the database to release 1 (9.0.1).

Important: Do not click **Migrate an Existing Database** because the Oracle Data Migration Assistant does not support Oracle9*i* Real Application Clusters. Instead, use the Migration utility to migrate after installation as described in the document *Oracle9i Database Migration*.

Note: Migrating or upgrading on Windows NT and Windows 2000 involves disabling the OracleCMService and starting and shutting down the **operating system-dependent clusterware (OSD)** layer at a specific point during migration. These instructions are not covered in the *Oracle9i Database Migration* guide. See the *Oracle9i Database Administrator's Guide for Windows* for complete information about these migration and upgrade steps.

Co-Existence of Oracle Versions on the Same Cluster

Whether different versions of Oracle can exist on the same cluster depend 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 Windows NT and Windows 2000, 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 you cannot have different versions of Oracle older than release 8.1 on the same cluster. For example, a release 8.0 and a release 8.1 Real Application Clusters database cannot co-exist on the same cluster.

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

For Windows NT and Windows 2000, operating system-dependent (OSD) layers from release 1 (9.0.1) 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 Oracle9*i* Real Application Clusters one at a time, without stopping the entire database. Release 1 (9.0.1) of Oracle9*i* Real Application Clusters does not support rolling upgrades.

Multiple Oracle Homes

Oracle9*i* Real Application Clusters on UNIX and Windows NT and Windows 2000 supports multiple Oracle homes, just as an **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 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 `libskgxp.so` to reside in the `$ORACLE_HOME/lib` directory where Oracle was installed.

5 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 Multi-Instance Environments
- Database Objects to Support Multiple Instances
- Creating a Database Manually

Setting CREATE DATABASE Options for Multi-Instance 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 which 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 option of CREATE DATABASE specifies the maximum number of **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, you should 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.

Setting MAXDATAFILES

The MAXDATAFILES option is generic, but Real Application Clusters databases tend to have more **datafiles** and log files than standard systems.

> **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 5-7 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 Multiple Instances

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

- Creating Additional Rollback Segments (Optional)
- Configuring the Online Redo Log for Real Application Clusters

Creating Additional Rollback Segments (Optional)

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 Rollback Segment Undo 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 contention, create these rollback segments in a tablespace other than the SYSTEM 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

Using Private Rollback Segments

If you use Rollback Segment Undo mode 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. This reserves the rollback segment 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, 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 multiple instances.

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 initialization parameter `THREAD` assigns a unique thread number to the 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 do 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, 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. Although rarely necessary, 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

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 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 Persistent Initialization Parameter File

Task 7: Create the Password Files

Task 8: Prepare a Cluster CREATE DATABASE Script

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 some existing files. Backup 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 Settings

In Real Application Clusters, each **node** has its own 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.

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 ID** 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 Example Settings for Database Name, Domain, Global Name, and sid Prefix

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

3. 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	The node name defined by the **Cluster Manager (CM)** software Use the command `lsnodes -l -n` to obtain the computer's node name. `lsnodes` is located in the `$ORACLE_HOME/bin` directory on UNIX and `%ORACLE_HOME%\bin` directory on Windows NT and Windows 2000.
Host name	The host name of the computer. The host name may be the same name as the node name. On UNIX and Windows NT and Windows 2000, use the command `hostname` to obtain the host name for the computer.
Thread ID	Each instance requires a unique **thread ID**. 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`.

4. 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
node1	node1	1	db1
node2	node2	2	db2

Task 3: Create the Real Application Clusters Configuration with SRVCTL

If this is the first Oracle9*i* database created on this **cluster**, 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 all nodes 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:

```
srvctl add db -p db_name -o oracle_home
```

Then for each instance enter the command:

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

Task 4: Configure the oratab File on UNIX

To use **Oracle Enterprise Manager**, manually create an entry in `oratab` file on each node. This entry identifies the database. Oracle Enterprise Manager uses this file during **service discovery** to determine the `ORACLE_HOME` from which it 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 the database should not be started at reboot time:

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

Use the database name and Oracle home you specified in Task 2: Specify the Database and Instance Settings

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**. To simplify administration, Oracle Corporation recommends that you use `sids` that consist of the database name as the common base and the thread ID of the node's instance that you specified in Task 2: Specify the Database and Instance Settings. For example, if `db` is the database name, 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:

- UNIX
- Windows NT and Windows 2000

UNIX On UNIX, you must set the `ORACLE_`*`sid`* environment variable.

> **See Also:** *Oracle8i Administrator's Reference* for your UNIX operating system systems for further information about setting this environment variable

Windows NT and Windows 2000 On Windows NT and Windows 2000, set the `ORACLE_`*`sid`* registry value in:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOMEID
```

> **See Also:** *Oracle9i Database Administrator's Guide for Windows* for further information about this registry value

After creating the *`sid`*`s`, create an OracleService*sid* service. You can use this service to start or stop an instance from the Control Panel. To create OracleService*sid*:

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

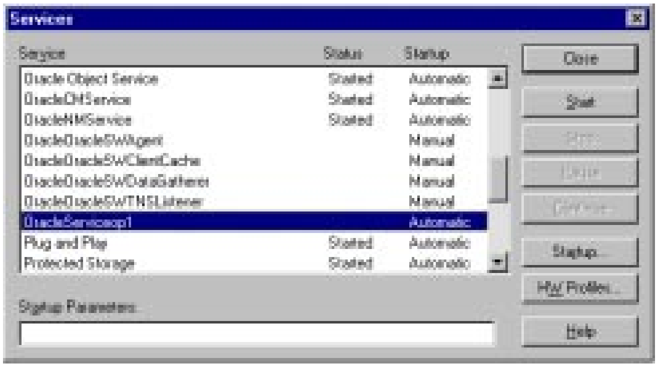
   ```
   C:\%ORACLE_HOME%\bin\ crtsrv sid
   ```

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

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

2. Verify OracleService*sid* exists by choosing the Services icon from the Control Panel. The Services screen appears as shown in Figure 5–1:

Figure 5–1 Windows NT and Windows 2000 Services Screen



Service	Status	Startup
Oracle Object Service	Started	Automatic
OracleCMService	Started	Automatic
OracleNMService	Started	Automatic
OracleOracleSWAgent		Manual
OracleOracleSWClientCache		Manual
OracleOracleSWDataGatherer		Manual
OracleOracleSWTNSListener		Manual
OracleServiceop1		Automatic
Plug and Play	Started	Automatic
Protected Storage	Started	Automatic

Task 6: Create the Persistent Initialization Parameter File

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

Name the server parameter file **initdb_name.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 in the `%ORACLE_HOME%\srvm\admin` directory on Windows NT and Windows 2000. Rename these files and edit and customize them for your database.

Edit the following parameters in the new `init`*`db_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

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 8-8 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 Oracle9*i* 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 orapw*sid* on UNIX and pwd*sid*.ora on Windows NT and Windows 2000, where *sid* identifies the database instance you specified in Task 2: Specify the Database and Instance Settings.

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 Cluster CREATE DATABASE Script

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` so it points to the location of the `init`*`db_name`*`.ora` file.

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. You must also be authorized to connect as **SYSDBA**.
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 Settings.
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, create enough private rollback segments for the number of concurrent users for each transaction. Oracle requires 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;
```

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
`listener.ora`	Includes addresses of each network **listener** on a server, the *sid*s of the databases for which they listen, and various control parameters used by the listener.	The `listener.ora` file on each node must be configured with: - Listener name - A TCP/IP address for Oracle Enterprise Manager - An entry for the *sid* of the **instance** in the *sid*_LIST_*listener_name* section **See Also:** - "Dedicated Server Configuration" on page 8-9 for a sample configuration file - "Configuring the Listener" in the *Oracle Net Services Administrator's Guide* for configuration procedures
`tnsnames.ora`	Includes a list of network descriptions of each **service name**, called **net service names**.	**See Also:** - "Net Service Names (tnsnames.ora File)" on page 8-13 for net service name requirements - "Configuring Naming Methods" in the *Oracle Net Services Administrator's Guide* for configuration procedures
`sqlnet.ora`	Includes the names resolution method.	Because the net service names are specified in `tnsnames.ora` files, the `sqlnet.ora` file must specify that the `tnsnames.ora` file be used when resolving a net service name. **See Also:** "Profile (sqlnet.ora File)" on page 8-17 for a sample configuration

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

6

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:

> **See Also:** *Oracle9i Real Application Clusters Administration* for more information on 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 as a result of self-tuning are not preserved after shutdown.

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

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 `PFILE`s 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 parameters 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/initsid.ora
```

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

```
%ORACLE_HOME%\database\initsid.ora
```

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

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

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

```
SPFILE='db_spfile1'
```

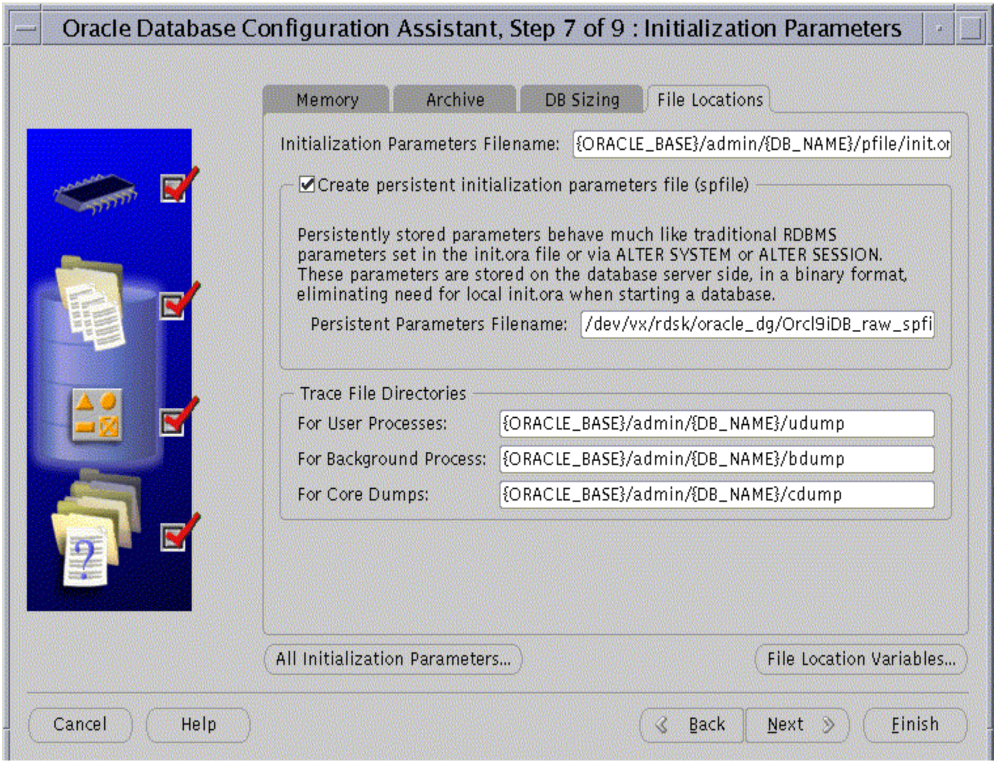
However, if your platform supports a cluster file system, use an alternate file location of:

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

You must use the correct location so that all instances use the same the 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 and enter the **raw device** path name in the `Persistent Parameters Filename` field as shown in Figure 6–1.

Figure 6–1 File Locations Tab For Initialization Parameters



Oracle Database Configuration Assistant, Step 7 of 9 : Initialization Parameters

Initialization Parameters Filename: {ORACLE_BASE}/admin/{DB_NAME}/pfile/init.or

☑ Create persistent initialization parameters file (spfile)

Persistently stored parameters behave much like traditional RDBMS parameters set in the init.ora file or via ALTER SYSTEM or ALTER SESSION. These parameters are stored on the database server side, in a binary format, eliminating need for local init.ora when starting a database.

Persistent Parameters Filename: /dev/vx/rdsk/oracle_dg/Orcl9iDB_raw_spfi

Trace File Directories

For User Processes: {ORACLE_BASE}/admin/{DB_NAME}/udump

For Background Process: {ORACLE_BASE}/admin/{DB_NAME}/bdump

For Core Dumps: {ORACLE_BASE}/admin/{DB_NAME}/cdump

Note: When you use the DBCA to create the server parameter file, the `PFILE` file name is `$ORACLE_HOME/dbs/init`*`sid`*`.ora` on UNIX and `%ORACLE_HOME%\database\init`*`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, on UNIX platforms for example, by examining directories in the following order:

1. `$ORACLE_HOME/dbs/spfile`*`sid`*`.ora`
2. `$ORACLE_HOME/dbs/spfile.ora`
3. `$ORACLE_HOME/dbs/init`*`sid`*`.ora`

The search order on Windows NT and Windows 2000 is:

1. `%ORACLE_HOME%\database\spfile`*`sid`*`.ora`
2. `%ORACLE_HOME%\database\spfile.ora`
3. `%ORACLE_HOME%\database\init`*`sid`*`.ora`

Migrating to the Server Parameter File in Real Application Clusters Environments

If you 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** multi-**instance** configurations, or if you are using a cluster file system, you can 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 example is for UNIX only.

1. Combine the **initialization parameter files** for all instances into a single `init`*`dbname`*`.ora` file by copying all shared IFILE contents *as is*. All the 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 `INIT`*`sid`*`.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/rdsk/oracle_dg/db_spfile'
FROM PFILE='?/dbs/initdb_name.ora'
```

This statement reads your combined `init`*`db_name`*`.ora` file that you created by merging your IFILEs and transfers the settings for parameters from the merged file into your the 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 `init`*`sid`*`.ora` contains the entry:

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

If you use this `STARTUP` command syntax, Oracle uses the server parameter file entry specified in `init`*`sid`*`.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 inaccurate parameter changes that you attempt using the `ALTER SYSTEM SET` statement. Oracle does this when an error occurs while reading 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 in case you need them for recovery purposes. Do this using the CREATE PFILE statement. For example:

```
CREATE PFILE='?/dbs/initdb_name.ora'
FROM SPFILE='/dev/vx/rdsk/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

Part Two 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 Two is:

- Chapter 7, "Installation and Configuration Highlights for Oracle Enterprise Manager in Real Application Clusters"

7

Installation and Configuration Highlights for Oracle Enterprise Manager in Real Application Clusters

Oracle Enterprise Manager supports the administration of **cluster** databases with **Server Management (SRVM)**. SRVM was previously known as *Oracle Parallel Server Management* (OPSM). SRVM enables the clustered components of Oracle Enterprise Manager. A performance monitoring tool called **Oracle Performance Manager** further enhances SRVM. It enables you to monitor cluster database performance using the global V$ view tables.

This chapter describes how to install and configure SRVM.

Specific topics discussed are:

- Server Management Architecture
- SRVM Requirements
- Understanding the Oracle Enterprise Manager Setup
- Install and Configure Oracle Enterprise Manager
- Configuring Oracle Performance Manager
- Additional Notes for Running the Console in Stand-Alone Mode

Server Management Architecture

You can use Server Management (SRVM) to control the activity of **Real Application Clusters** and their instances. SRVM is a comprehensive, integrated system management solution for a Real Application Clusters. SRVM enables you to manage cluster databases through an open client-server architecture.

Oracle Enterprise Manager provides database administrators (DBAs) tools to manage, monitor, and administer even the most complex network of databases from a single workstation, called the Enterprise Manager Console. Oracle Enterprise Manager can be used with single instance databases as well as with Real Application Clusters.

> **See Also:** *Oracle9i Real Application Clusters Administration* for further information about using views not available with Oracle Enterprise Manager, and for more detail on monitoring and tuning Real Application Clusters

Within the context of Oracle Enterprise Manager, SRVM contains the utilities shown in Table 7–1 to manage instances:

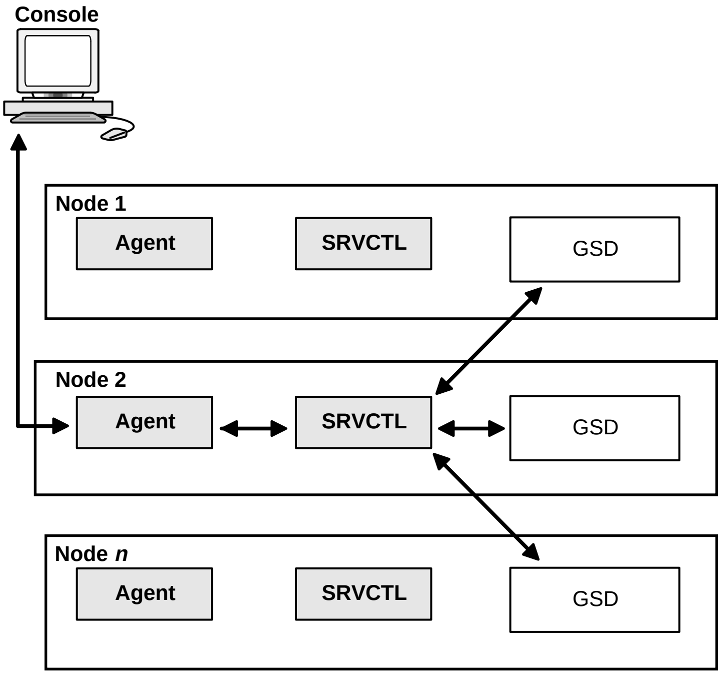
Table 7–1 SRVM Utilities

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

The Global Services Daemon (GSD) background process allows you to use Oracle Enterprise Manager 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 GSD then receives requests from SRVCTL, 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 manually

SRVM Requirements

The requirements for each node that you must meet to use SRVM from the **Console** are shown in Table 7–2. If you followed the database creation procedures as described earlier in this book, these requirements should have been satisfied.

Table 7–2 Node Requirements

Node Requirements	References
Ensure **Oracle9i Enterprise Edition**, Real Application Clusters Option, and Oracle Intelligent Agent are installed on each node in the cluster.	Part One of this book, "Installing Oracle9i Real Application Clusters".
Each instance's Oracle System Identifiers (**sid**) in the cluster database must be unique.	Part One of this book, "Installing Oracle9i Real Application Clusters".
Each node must have a `tnsnames.ora` file configured with a **net service name** entry for: ■ The database ■ Each instance	Part Three of this book, "Managing the Configuration".
The **listener.ora** file must define a **listener** for each node with: ■ A TCP/IP address for the listener ■ An entry for the *`sid`* of the node's instances in the *`sid`*`_LIST_listener_name` section.	Part Three of this book, "Managing the Configuration".
The `oratab` file must be configured on UNIX for the agent to discover the database.	Part One of this book, "Installing Oracle9i Real Application Clusters".
For UNIX, Windows NT, and Windows 2000, ensure GSD has been started on each managed node by the *oracle* account at system startup. GSD resides in `$ORACLE_HOME/bin` on UNIX and in `%ORACLE_HOME%\bin` on Windows NT and Windows 2000.	Part One of this book, "Installing Oracle9i Real Application Clusters". This part describes how to run the command `gsd` in the `ORACLE_HOME/bin` directory.
The `srvcfg` cluster configuration file must have entries for the cluster database and instances.	Part One of this book, "Installing Oracle9i Real Application Clusters".

Understanding the Oracle Enterprise Manager Setup

You can run the individual Oracle Enterprise Manager components on separate machines or combine different components on separate machines to collaboratively manage the complete Oracle environment. Two environments are described under the following headings "Console, Management Server, and Repository on Same Machine" and "Console, Management Server, and Repository on Separate Machines".

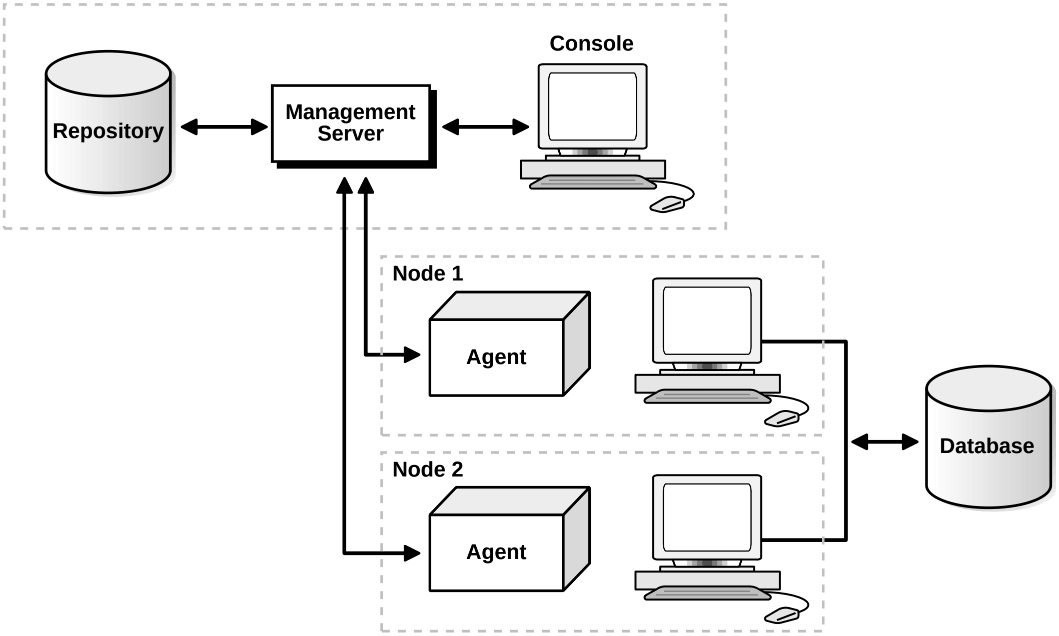
Console, Management Server, and Repository on Same Machine

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 Windows NT, Windows 2000, or Solaris machine with an Oracle database installed that is only used as a repository. Oracle Intelligent Agent does *not* need to be running.

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

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



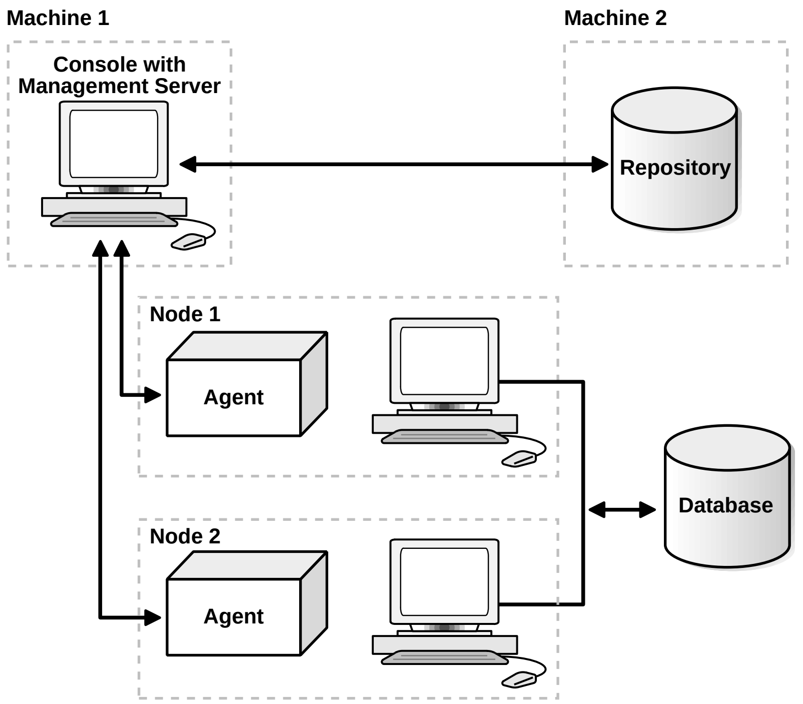
Console, Management Server, and Repository on Separate Machines

The Repository is on a separate Solaris, Windows NT, or Windows 2000 machine with an Oracle database installed that is used solely as a repository. See Figure 7–3. Oracle Intelligent Agent does *not* need to be running.

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 Oracle Intelligent Agent is running on *both* nodes.

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



Install and Configure Oracle Enterprise Manager

The installation and configuration process for Oracle Enterprise Manager is described in this section.

Task 1: Oracle Enterprise Manager Installation

1. Install Oracle Enterprise Manager. Note that you can either install a complete Oracle9*i* database or Oracle9*i* Management and Integration. (Oracle Enterprise Manager 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 Oracle Enterprise Manager.
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 Oracle Enterprise Manager configuration instructions.
- *Oracle8i Installation Guide* for Compaq Tru64, Hewlett-Packard HPUX, IBM AIX, Linux, and Sun Solaris-based systems, and *Oracle9i Database installation guide for Windows* to install Oracle Enterprise Manager without Oracle Diagnostics Pack
- *Oracle Diagnostics Pack Installation* to install Oracle Enterprise Manager and Oracle Diagnostics Pack

Task 2: Specify Preferred Credentials for Nodes and Database

You must configure the Oracle Enterprise Manager with preferred user credentials so you can perform certain functions. Oracle Enterprise Manager uses these credentials when establishing connections to Real Application Clusters, such as when the database is expanded in the Console's navigator. Startup and shutdown operations also use the credentials.

The 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, Oracle Enterprise Manager 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 accomplish a number of tasks related to creating accounts, granting privileges, and setting credentials. This section covers the following topics:

- Task 3: Create an Operating System Account
- Task 4: Grant SYSDBA or SYSOPER Privileges to a Database User

- Task 5: Set User Credentials in the Console

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, you must create a Windows user account. You must be a member of the "Administrators" group. 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 a user is identified, 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:

```
GRANT sysdba to username;
GRANT sysoper to username;
```

Task 5: Set User Credentials in the Console

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

To set credentials:

1. Choose System > Preferences.

   The Edit User Preferences dialog box appears.

2. Click the **Preferred Credentials** tab to enter credentials for various service types:

3. Click on a node (identified by the Node service type) and enter the operating system user name and password.

4. Perform Step 3 for each node in the cluster.

5. Click on the Real Application Clusters database (identified by the service 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 Real Application Clusters database and instance startup and shutdown.

---

6. To use the Oracle Performance Manager application, click on an instance (identified by the Instance service type), enter a user name and password that can connect to the instance, select **NORMAL** role from the drop-down menu.
7. Click **OK**.

See Also:

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

Configuring Oracle Performance Manager

Note: Oracle Enterprise Manager is *not* required by the Oracle Performance Manager because its functions are performed using a database connection to the cluster database. Oracle Performance Manager can be run with or without running Oracle Enterprise Manager. If you run this product as a stand-alone product, you do not have to configure Oracle Enterprise Manager.

To configure Oracle Performance Manager, perform these tasks:

Task 1: Start Oracle Performance Manager

Task 2: Accessing Statistical Charts

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.

Instances Folder

If the login occurred from the Console or the **Login to the Oracle Management Server** option was selected in the Performance Manager Login dialog box, you can expand the 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 referred to with Oracle Parallel Server names. When you are accessing a cluster database of version 9.0.1 and later, you will see charts referred to with Real Application Clusters 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 7-7, 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 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 reports
- *Getting Started with the Oracle Standard Management Pack* for general information about the Oracle Performance Manager application

Additional Notes for Running the Console in Stand-Alone Mode

To run the Oracle Enterprise Manager console in stand-alone mode without a connection to the Management Server, a `tnsnames.ora` file must be created. It must contain entries for the database and each instance.

When running the DBA Studio application in stand-alone mode, the Startup and Shutdown menu items start and stop an individual **instance**, not the Real Application Clusters database.

Part III

Managing the Configuration

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

- Chapter 8, "Understanding The Installed Configuration"
- Chapter 9, "Configuring High Availability Features for Real Application Clusters"

8

Understanding The Installed Configuration

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

- Understanding the Configured Environment
- Raw Device Configuration
- UNIX Operating System Configurations
- Database Components Created Using the Oracle Database Configuration Assistant
- Managing Undo Tablespaces
- Initialization Parameter Files
- Shared Server Configuration
- Dedicated Server Configuration
- 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 **Oracle Database Configuration Assistant (DBCA)** meet all the requirements for database creation and **Oracle Enterprise Manager** discovery of Real Application Cluster databases. The following sections describe the configured environment in detail:

- Raw Device Configuration
- UNIX Operating System Configurations
- Shared Server Configuration
- Dedicated Server Configuration
- Directory Server Access (ldap.ora File)
- Net Service Names (tnsnames.ora File)
- Profile (sqlnet.ora File)

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

Raw Device Configuration

The Oracle Database Configuration Assistant uses a shared **raw device** in UNIX environments, or a shared logical partition in Windows environments, to store Real Application Clusters database configuration information. The **Oracle Universal Installer (OUI)** automatically initializes the shared device by executing the `srvconfig -init` command. If this initialization does not complete, manually initialize the raw device by executing the `srvconfig -init` command.

You can also use the `srvconfig` command to import or export the contents of a configuration raw volume on UNIX, or the contents of a logical drive on Windows platforms, to or from a text file. You can also use `srvconfig` to convert a pre-Oracle9*i* Oracle Parallel Server *dbname*`.conf` file to a shared raw device. Refer to the following sections for platform-specific information on the `srvconfig` command.

Executing srvconfig on UNIX Platforms

When you execute `srvconfig -init,` this command syntax requires that the `srvConfig.loc` file exists and that it points to a raw device or shared cluster file system file.

Using the `path_name` as the complete path name for the raw device you specify, the `srvConfig.loc` file contains the entry:

```
srvconfig_loc=path_name
```

The `srvConfig.loc` file resides in:

- The `/var/opt/oracle` directory on Sun Solaris
- The `/etc` on directory HP-UX

Executing srvconfig on Windows

On Windows NT and Windows 2000, when you execute `srvconfig -init` you must have a symbolic link named `srvcfg` 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.

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 a Real Application Clusters database name as well whether it should be auto-started on reboot. 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 reboot time. A sample entry follows for a database named `db`:

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

Database Components Created Using the Oracle Database Configuration Assistant

This section describes the database components created by the Oracle 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- and multiple-**instance** environments is divided into smaller logical areas of space known as **tablespaces.** Each tablespace corresponds to one **datafile** 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 Oracle9*i* databases.
SPFILE	This is a tablespace for the binary server parameter file.
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/or 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 that the Oracle 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.

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

Tablespace Name	Contents
TOOLS	Stores tables for Oracle Enterprise Manager.
DRSYS	Consists of data for Oracle9*i* *inter*Media Text.
EXAMPLE	Stores the Oracle9*i* Sample Schemas
CWMLITE	Stores the OLAP files.

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 database file. 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

Tablespaces	Windows NT and Windows 2000 Symbolic Link Names
SYSTEM	*db_name*_system1
SPFILE	*db_name*_spfile1
USERS	*db_name*_users1
TEMP	*db_name*_temp1
UNDOTBS1	*db_name*_undotbs1
UNDOTBS2	*db_name*_undotbs2
RBS (optional)	*db_name*_rbs1
EXAMPLE	*db_name*_example1
CWMLITE	*db_name*_cwmlite1
INDX	*db_name*_indx1
TOOLS	*db_name*_tools1
DRSYS	*db_name*_drsys1
Control File 1	*db_name*_control1

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

Tablespaces	Windows NT and Windows 2000 Symbolic Link Names
Control File 2	*db_name*_control2
SRVM Configuration	srvcfg

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

Redo Log Files

Each instance is configured with two **redo log files** that are stored on the raw devices. The redo log files' names created with the preconfigured database configuration options vary by operating system. You must enter the file names on UNIX unless you are using a cluster file system.

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 Customize database creation method, locate the redo log files in the Storage Page and set their filenames fields to the correct raw devices or symlink names.

> **Note:** Where the notation *db_name* appears above 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 Rollback Segment Undo mode.

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 `rbs`*`thread_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** 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. Although not recommended, 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

Shared Server Configuration

If you use the Transaction Processing preconfigured database option, Oracle configures the database for **shared server**. For an Customized installation, shared server is configured if 20 or more users are configured. If you use the **New Database** database creation type, then you have a choice of using either shared server or dedicated server.

Shared server is configured as in the following example:

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

This configuration enables **connection load balancing.** Connection load balancing equalizes the number of active connections among the various instances and shared server **dispatchers** for the same service. The `(listener=listeners_`*`db_name`*`)` setting enables an instance to register its instance load information with remote listeners on the other nodes. The TNS **connect descriptor** to which `listeners_db_name` refers should include the addresses of each **listener** that listens for instances of this cluster database.

Oracle resolves `listeners_db_name` to listener addresses through a `tnsnames.ora` file like the one shown in Example 8–1 on page 8-16. In shared server configurations, a listener selects a dispatcher in the following order: 1) least loaded node, 2) least loaded instance, and 3) least loaded dispatcher for that instance.

See Also:

- "Net Service Names (tnsnames.ora File)" on page 8-13
- The *Oracle Net Services Administrator's Guide* for further information about the `DISPATCHERS` parameter

Dedicated Server Configuration

A **dedicated server** configuration requires one dedicated server process for each user process. You can implement connection load balancing in Real Application Clusters, however, the throughput and scalability of dedicated server processing is limited by the ability of the server processes to accommodate the user load.

When using dedicated server, configure connection load balancing as in the following example:

```
REMOTE_LISTENERS=listeners_db_name
```

This configuration also enables **connection load balancing.** Connection load balancing equalizes the number of active connections among the various instances for the same service. The `REMOTE_LISTENERS` setting enables an instance to register its instance load information with remote listeners on the other nodes. The TNS connect descriptor to which `listeners_db_name` refers should include the addresses of all listeners that listen for instances of this cluster database.

Oracle resolves `listeners_db_name` to listener addresses through a `tnsnames.ora` file like the one shown in Example 8–1 on page 8-16. In dedicated server configurations, a listener selects an instance in the following order: 1) least loaded node, and 2) least loaded instance.

See Also: *Oracle Net Services Administrator's Guide* for further information about 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`, the Oracle 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, 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 Oracle9*i* release 1 (9.0.1) 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=/orahome81)
      (program=extproc)
     (sid_desc=
       (oracle_home=/orahome81)
       (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=/u01/app/oracle/9.0.1)))
```

> **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:** *Oracle 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 an 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:** *Oracle 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.

Oracle 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	Oracle Enterprise Manager searches for a net service name entry for the database. This entry enables Oracle Enterprise Manager to discover a Real Application Clusters database and to determine which instances to use for a connection. A listener protocol address is configured for each instance. In addition, the LOAD_BALANCE and FAILOVER options force the address to be chosen randomly. If the chosen address fails, the connection request is failed over to the next address. This way, if an instance should go down, Oracle Enterprise Manager can still connect by way of another instance. In the following example, db.us.acme.com is used by Oracle Enterprise Manager to connect to the target database, db.us.acme.com. 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))) **Note:** FAILOVER=ON is set by default for a list of addresses. Thus, you do not need to explicitly specify the FAILOVER_ON parameter.

Instance connections	Oracle Enterprise Manager searches for a net service name entry for each instance. This entry enables Oracle Enterprise Manager to discover the instances in the cluster. These entries are also used to start and stop instances.
	In the following example, db1.us.acme.com, is used by Oracle Enterprise Manager to connect to an instance named db1 on db1-server:
	db1.us.acme.com=
	 (description=
	  (address=(protocol=tcp)(host=db1-server)(port=1521))
	  (connect_data=
	    (service_name=db.us.acme.com)
	    (instance_name=db1)))

Table 8–3 Connections for Net Service Names

Net Service Name Type	Description

Remote listeners	As discussed in "Shared Server Configuration" on page 8-8, the `DISPATCHERS` parameter is set in the `init`*`sid`*`.ora` file with the `LISTENER` attribute:
	`DISPATCHERS="(protocol=tcp)(listener=listeners_`*`db_name`*`)"`
	This enables the instance to know about the remote listeners on the other nodes. `listeners_`*`db_name`* is then resolved through the `tnsnames.ora` file.
	In the following example, `listeners_db.us.acme.com` is resolved to list of listeners available in the cluster database as shown below:
	`listeners_db.us.acme.com=`
	`   (address=(protocol=tcp)(host=db1-server)(port= 1521))`
	`   (address=(protocol=tcp)(host=db2-server)(port=1521))`
	The instance uses this list to determine the names of the remote listeners to register its information.

Nondefault listeners	As discussed in "Nondefault Listeners" on page 8-10 and "Multiple Listeners" on page 8-10, the `LOCAL_LISTENER` parameter is set in the `init`*`sid`*`.ora` file if a nondefault listener is configured.
	`sid``.local_listener=listener_`*`sid`*
	`listener_`*`sid`* is resolved to a listener address.
	In the sample below, `listener_db1.us.acme.com` is resolved to the nondefault listener address:
	`listener_db1.us.acme.com=`
	`   (address=(protocol=tcp)(host=db1-server)(port= 1421))`

External procedures	An entry for connections to external procedures. This enables an Oracle9*i* database to connect to external procedures.
	`extproc_connection_data.us.acme.com=`
	` (description=`
	`  (address_list=`
	`    (address=(protocol=ipc)(key=extproc0))`
	`  (connect_data=`
	`    (sid=plsextproc)))`

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:** *Oracle 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 *Oracle Net Services Administrator's Guide* for further information about the `sqlnet.ora` file

9

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 and Secondary Instances
- 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:** *Oracle Net Services Administrator's Guide* for detailed information about the sub-parameters you can use for FAILOVER_MODE and the *Oracle Real Application Clusters Guard Administration and Reference Guide* 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, OracleNet 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)
       (port=1521))
 (address=
       (protocol=tcp)
       (host=db2)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
      (failover_mode=
        (type=select)
        (method=basic))))
```

See Also: *Oracle 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. If the initial connection fails, it fails over to addresses in the description lists.

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

> **See Also:** *Oracle 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. If db1 fails after the connection, Oracle Net fails over to db2, 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)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (instance_name=db1)
     (failover_mode=
       (backup=db2.acme.com)
       (type=select)
       (method=preconnect))))
db2.acme.com=
 (description=
  (address=
       (protocol=tcp)
       (host=db2)
       (port=1521))
  (connect_data=
     (service_name=db.us.acme.com)
     (instance_name=db2)
     (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 and Secondary Instances

The primary and 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 File 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`*`sid`*`.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 File Configuration

To enable primary and secondary instance configuration, configure the initialization 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)(port=1521))
  (address=(protocol=tcp)(host=db2)(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:

- *Oracle Net Services Administrator's Guide* to configure a connect descriptor
- *Oracle 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
`listener=`	`listener=`
` (description=`	` (description=`
`  (address=`	`  (address=`
`   (protocol=tcp)`	`   (protocol=tcp)`
`   (host=db1)`	`   (host=db1)`
`   (port=1521)))`	`   (port=1521)))`
***sid*_list_listener=**	
(*sid*_desc=	
(oracle_home=/orahome81)	
(*sid*_name=db1))	

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:** *Oracle 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 the performance of the Oracle database server. In Real Application Clusters primary and secondary instance configurations, the library cache associated with the primary instance contains up-to-date information. If failover occurs, then the benefit of that information is lost unless the library cache on the secondary instance is populated beforehand.

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:** *Oracle Real Application Clusters Guard Administration and Reference Guide* for more information about this feature and *Oracle9i Supplied PL/SQL Packages and Types Reference* for more information about using DBMS_LIBCACHE

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)(port=1521))
  (address=(protocol=tcp)(host=db2)(port=1521))
  (connect_data=
     (service_name=db.us.acme.com)))
```

See Also:

- *Oracle Net Services Administrator's Guide* to configure a connect descriptor
- *Oracle 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

Migration

Part Four provides information on migrating to Real Application Clusters. The chapter in Part Four is:

- Chapter 10, "Migrating to Real Application Clusters"

10

Migrating to Real Application Clusters

This chapter describes procedures for migrating from single **instance** to multi-instance, **Real Application Clusters** databases. The topics in this chapter are:

- Moving from a Single Instance to Real Application Clusters
- Migrating to Real Application Clusters When Using Raw Devices or Shared File Systems
- Migrating Oracle8i Configurations to Oracle9i

Moving from a Single Instance to Real Application Clusters

This section explains how to enable your single instance database to support multiple instances. It also explains how to begin a project with a single instance Oracle database even though you intend to migrate to multi-instance Real Application Clusters. This section covers the following topics:

- Deciding to Migrate
- Preparing to Migrate
- Migrating the Database from Single to Multi-Instance

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

Deciding to Migrate

You may decide to migrate to a multi-instance database when you have designed your application with Real Application Clusters in mind. However, 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** processing

You can 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. You can also use Oracle with a clustered file system.

Preparing to Migrate

To migrate to Real Application Clusters, you must consider both the hardware and software requirements as described in this section, and the administrative issues of migrating applications to cluster database environments.

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 Oracle9*i* Real Application Clusters

Administrative Issues of Migrating Your Application from Single to Multi-Instance

Note the following administrative issues of conversion:

- Your backup procedures should be in place before migrating from single instance Oracle to Real Application Clusters.
- Additional archiving considerations apply in Real Application Clusters environments. In particular, the archive file format needs the thread number. Furthermore, archived logs from all nodes are needed for media recovery. If you archive to a file, a method of accessing the archive logs is required where file systems are not shared.

Migrating the Database from Single to Multi-Instance

This procedure explains how to migrate a database from single instance to multi-instance 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 to Real Application Clusters When Using Raw Devices or Shared File Systems" on page 10-6.

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: 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.

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 Oracle9*i* Enterprise Edition and Oracle9*i* Real Application Clusters

Except on the server already running the **Oracle9i Enterprise Edition**, install Oracle9*i* Enterprise Edition along with the Oracle9*i* Real Application Clusters option (formerly known as *Oracle Parallel Server*), as described in *Oracle9i Real Application Clusters Installation 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 Oracle9*i* Enterprise Edition, install Oracle9*i* 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 **Oracle 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 Installation 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 Guide and Reference* for more details about optimizing your memory configuration

Task 9: Start the Database

For more information on this procedure, refer to *Oracle9i Real Application Clusters Administration*.

Migrating to Real Application Clusters When Using 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, you do not have to rebuild the database as described in the previous set of 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 the Database from Single to Multi-Instance" 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 migrating from 7.3, 8.0, and 8i to 9.0.1

Migrating Oracle8*i* Configurations to Oracle9*i*

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

1. Stop all **Global Services Daemon (GSD)**.
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 Oracle8*i* environment variable.

> **Note :** Environment variables are case-sensitive.

Part V

Reference

Part Five provides reference information for Real Application Clusters. The contents of Part Five are:

- Appendix A, "Directory Structure for Real Application Clusters Environments"
- Appendix B, "Oracle Enterprise Manager in Real Application Clusters Reference"
- 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 Structure for Real Application Clusters
- Windows NT and Windows 2000 Directory Structure for Real Application Clusters

Understanding the Real Application Clusters Directory Structure

When you install **Oracle9i Enterprise Edition** and the Oracle9*i* Real Application Clusters Option, all subdirectories are under a top-level ORACLE_BASE. ORACLE_HOME and admin directories are also located under ORACLE_BASE.

UNIX Directory Structure for Real Application Clusters

The following is the hierarchical directory tree of a sample OFA-compliant database for Real Application Clusters on UNIX platforms:

$ORACLE_BASE	/u01/app/oracle is the default ORACLE_BASE directory
$ORACLE_HOME	/product/9.0.1 is the name of the Oracle home by default
/bin	Subtree for Oracle binaries
/network	Subtree for Oracle Net

	/srvm		Server Management message and TCL files
		/admin	clustdb.sql script and initialization parameter files for database creation
	/dbs		This is a legacy directory from previous releases. It contains links to **initialization parameter files** that point to the new location files, $ORACLE_BASE/admin/db_name/pfile.
	...		
/admin			Subtree for Real Application Clusters database administration files
	/db_name		Database administration files for this database identified by the database name
		/adhoc	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

See Also: *Oracle9i Administrator's Reference* for your UNIX operating system for further information about $ORACLE_HOME and /admin directories

Windows NT and Windows 2000 Directory Structure for Real Application Clusters

The following is the hierarchical directory tree of a sample OFA-compliant database for Real Application Clusters on Windows NT and Windows 2000:

x:\oracle_base				c:\oracle is the default ORACLE_BASE directory.
	\%ORACLE_HOME%			\ora9.0.1 is the name of the Oracle home by default.
		\bin		Subtree for Oracle binaries.
		\network		Subtree for Oracle Net configuration files, including tnsnames.ora, **listener.ora** and sqlnet.ora.
		\srvm		\admin subdirectory.
			\admin	The Real Application Clusters script clustdb.sql 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, ORACLE_BASE\admin\db_name\pfile.

	...		
\admin			Subtree for Real Application Clusters database administration files
	\db_name		*db_name* database administration files for the **instance** identified by *sid*.
		\adhoc	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

See Also: *Oracle9i Database installation guide for Windows* for further information about the contents of the *ORACLE_HOME* and \admin directories

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 `lsnrctl dbsnmp_start` command. Ensure that the DBSNMP utility is set up to run at system startup time.

B

Oracle Enterprise Manager in Real Application Clusters Reference

Appendix B describes **service discovery** problems for **Oracle Enterprise Manager** in **Real Application Clusters** environments. Specific topics covered in this appendix are:

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 was not starting on the **node** or the configuration is incorrect. If starting the Oracle Intelligent Agent does not resolve the problem, the discovery failure could be due to a more serious configuration issue.

This section covers the following topics:

- Understanding Discovery
- Discovery Results
- Troubleshooting Discovery

Understanding Discovery

To understand proper configuration, it is important to understand how discovery works. 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. This file contains information about the nodes and their services (databases, instances and listeners) discovered.

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.

This 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:

- oratab on UNIX and Registry on Windows NT/Windows 2000
- listener.ora
- tnsnames.ora

You must accurately configure each of these components in order for discovery to work correctly.

Note: The following Discovery descriptions apply only to Release 9.0.1 and later. If you are running an earlier version, please refer to the corresponding editions of Real Application Clusters documentation.

oratab on UNIX and Registry on Windows NT/Windows 2000

Discovery first discovers the Real Application Clusters database name and the nodes associated with the database. How it accomplishes this depends on whether the managed system is running on:

- UNIX
- listener.ora

UNIX On UNIX operating systems, discovery uses information in the `oratab` entry for the name of the cluster. The `oratab` entry is found in `/etc/oratab` or `/var/opt/oracle/oratab`. It 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 runs:

```
srvctl config -p db_name
```

where *db_name* was retrieved from oratab,

On some operating systems, such as Sun Solaris, `node_list` defaults to the entire cluster, and this parameter does not need to be explicitly set.

The configuration file must exist even if it has no entries.

listener.ora Discovery locates the **listener** and **instance** names for a node with 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 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=rachp1-pc)(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.

tnsnames.ora

The `tnsnames.ora` file, 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_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)))
```

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 example below shows a database named db.us.acme.com running on node db1-server. The database address comes from the db.us.acme.com **net service name** in the tnsnames.ora file.

  ```
  db.us.acme.com=(rac_database, db1-server, (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)))), 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
  - Address of instance obtained from tnsnames.ora file
  - Real Application Clusters database name to which the instance belongs
  - Listener name

  The example below 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,
  listener_db1-server)
  ```

- 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:** Oracle Intelligent Agent must be running, and the Global Services Daemon (GSD) must be started before Discovery will function properly. Also note that the Agent must be in the same Oracle Home in which the database resides.

If the `services.ora` file contains an `ORACLE_DATABASE` entry instead of `rac_database` and `ops_instance` entries, discovery has failed. To resolve this:

1. Check that the database is defined correctly:

   On UNIX:

   a. Verify that `oratab` file is configured correctly.

   b. Run the following command to verify proper setup:

   ```
   srvctl config -p db_name -n node_name
   ```

   On Windows NT or Windows 2000:

   a. Check the registry entries associated with the database.

   b. Run the following command to verify proper setup:

   ```
   SRVCTL config -p db_name -n node
   ```

   On Unix, Windows NT, or Windows 2000, SRVCTL will display the name of the node and the instance for the node. The following example shows a node named NODE1 running an instance named DB1 with a listener named `LISTENER_NODE1`.

   ```
   node1 op1
   ```

2. 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 Worldwide Customer Support

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.

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, each typically running on different nodes, that coordinate with one another when accessing the shared database residing on disk.

Cluster Manager (CM)

An operating system-dependent component that discovers and tracks the membership state of nodes by providing a common view of cluster membership across the cluster.

CM monitors process health, specifically the health of the database instance. The **LMON**, a background process that monitors the health of the **Global Cache Service (GCS)**, registers and de-registers from CM.

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

The destination service is indicated by using its **service name** for Oracle release 8.1 databases or greater or its **Oracle system identifier (sid)** for Oracle release 8.0 or version 7 databases. The network route provides, at a minimum, the location of the listener through use of a network address.

connect identifier

A **net service name** or **service name**, that resolves to a **connect descriptor**. Users initiate a connect request by passing a user name and password along with a connect identifier in a connect string for the service to which they wish to connect, for example:

```
CONNECT username/password@connect_identifier
```

connection load balancing

A feature that balances the number of active connections among various instances and **shared server dispatchers** for the same service. Because of service registration's ability to register with remote listeners, a listener is always aware of all instances and dispatchers regardless. This way, a listener can sends an incoming client request for a specific service to the least loaded instance and least loaded dispatcher regardless of its location.

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.

control file

A file that records the physical structure of a database and contains the database name, the names and locations of associated databases and online redo log files, the timestamp of the database creation, the current log sequence number, and checkpoint information.

Customized

The Oracle 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.

datafile

File that contain the contents of logical database structures, such as tables and indexes. One or more datafiles form a logical unit of storage called a tablespace. A datafile can be associated with only one tablespace, and only one database.

Data Warehouse

The Oracle 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. Oracle Net sends the address of an existing server process back to the client. The client then resends its connect request to the server address provided. Contrast with **shared server**.

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 Oracle 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)

Real Application Clusters software that provides mechanisms to control the allocation and modification of Oracle resources.

Global Cache Service Processes (LMSn)

Global Cache Service Processes are processes that, when spawned by Oracle, transmit both the consistent read and the current blocks from holding instances to requesting instances. Real Application Clusters software provides for up to 10 LMS processes; however, there is by default one LMS processes per pair of CPUs. In general, the number of LMS processes varies depending on the amount of messaging traffic among nodes 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`.

The database name portion, in this case `sales`, should be an easy to remember name that you wish to call your database. The database domain portion, `us.acme.com`, specifies the database domain in which the database is located, making the global database name unique. When possible, Oracle Corporation recommends that your database domain mirror the network domain.

The global database name is the default service name of database, as specified by the `SERVICE_NAMES` parameter in the common database initialization file, **initdb_name.ora**.

Global Enqueue Service Monitor (LMON)

A process that handles remote resource requests. Remote requests are requests that originate from another instance.

Global Services Daemon (GSD)

Formerly OPSD. The Global Services Daemon is 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. GSD is installed on all nodes; do not delete the GSD process.

GSD

See **Global Services Daemon (GSD)**.

hybrid

A hybrid database is one that has both online transaction and data warehouse processing characteristics.

initialization parameter file

Files that contains information to initialize the database (**initdb_name.ora**) and instances (**initsid.ora**).

init*sid*.ora

An instance initialization file that contains parameters unique for an instance and points to **initdb_name.ora** for database parameters.

init*db_name*.ora

A common database initialization file shared among the instance that contains database parameters.

Input/Output (I/O)

Input/Output is an operating system-dependent component that provides I/O to access shared disks. See *Oracle9i Real Application Clusters Concepts*

instance

For a Real Application Clusters database, each node within the cluster has an instance of the running Oracle9*i* software referencing the database.

When a database is started on a database server (regardless of the type of computer), Oracle allocates a memory area called the **System Global Area (SGA)** and starts one or more Oracle processes. This combination of the SGA and the Oracle processes is called an instance. The memory and processes of an instance efficiently manage the associated database's data and serve the database users. You can connect to any instance to access information within a Real Application Clusters database.

Each instance has a unique **Oracle system identifier (sid)**, **INSTANCE_NAME**, **INSTANCE_NUMBER**, **rollback segments**, and **thread ID**.

INSTANCE_NAME

Represents the name of the instance and is used to uniquely identify a specific instance when multiple instances 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. The instance cannot use data blocks in another free list unless the instance is restarted with that instance number.

You can use various SQL options with the `INSTANCE_NUMBER` initialization parameter to associate extents of data blocks with instances.

The instance number is depicted by the `INSTANCE_NUMBER` parameter in the instance initialization file, **initsid.ora**.

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 separate process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server.

The listener brokers the client request, handing off the request to the server. Every time a client (or server acting as a client) requests a network session with a server, a listener receives the actual request. If the client's information matches the listener's information, the listener grants a connection to the server.

listener.ora

A configuration file for the **listener** that identifies the:

- Listener name
- Protocol addresses that it is accepting connection requests on

- Services it is listening for

The `listener.ora` file typically resides in `$ORACLE_HOME/network/admin` on UNIX platforms and `%ORACLE_HOME%\network\admin` on Windows NT and Windows 2000.

An Oracle release 8.1 database or greater does not require identification of the database service because of **service registration**. However, static service configuration is required for Oracle release 8.1 databases or greater if you plan to use Oracle Enterprise Manager.

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 Enqueue Service Monitor (LMON)**.

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. High performance and scalability is ensured because the workload is automatically shared and balanced when there are multiple Management Servers.

multi-threaded server (MTS)

See **shared server**.

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 along with a net service name in a connect string for the service to which they wish to connect:

```
CONNECT username/password@net_service_name
```

Depending on your needs, net service names can be stored in a variety of places, including:

- Local configuration file, `tnsnames.ora`, on each client
- Directory server
- Oracle Names server
- Active Directory Service
- External naming service, such as Novell Directory Services (NDS), Network Information Service, or Cell Directory Service (CDS)

Net8

See **Oracle Net**.

Net8 Assistant

See **Oracle Net Manager**.

Net8 Configuration Assistant (Net8CA)

See **Oracle Net Configuration Assistant**.

New Database

The Oracle Database Configuration Assistant's template that does not include datafiles. This template is fully customizable.

node

A machine where 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 clusterware (OSD)

A software layer that consists of several software components developed by vendors. The OSD layer maps the key operating system / cluster-ware services required for proper operation of Real Application Clusters.

OPSCTL utility

See **SRVCTL Utility**.

Oracle9*i* Enterprise Edition

Oracle9*i* Enterprise Edition is an Object-Relational Database Management System (ORDBMS). It provides the applications and files to manage a database. All other Oracle9*i* Real Application Clusters components are layered on top of Oracle9*i* 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 reboot your system.

Oracle Parallel Server Communication Daemon (OPSD)

See **Global Services Daemon (GSD).**

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 Database Configuration Assistant (DBCA)

A GUI tool that you can use to create 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.

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.

A product family consists of system management tools designed to efficiently manage the complete Oracle environment.

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

The foundation of Oracle’s family of networking products. Oracle Net allows services and their applications to reside on different computers so they can communicate as peer applications. The main function of Oracle Net is to establish network sessions and transfer data between a client machine and a server or between two servers. Once a network session is established, Oracle Net acts as a data courier for the client and the server.

Oracle Net Manager

Oracle Net Manager is 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.

You can use Oracle Net Manager to configure the following network components:

- Naming: Define **connect identifiers** and map them to **connect descriptors** to identify the network location and identification of a service. Oracle Net Manager supports the configuration of connect descriptors in local `tnsnames.ora` files, centralized LDAP-compliant directory service, or an Oracle Names server.
- Naming Methods: Configure the different ways in which connect identifiers are resolved into connect descriptors.
- Listeners: Create and configure listeners to receive client connections.

Oracle Net Configuration Assistant

A post-installation tool that configures basic network components after installation, including:

- Listener names and protocol addresses
- Naming methods the client will use to resolve **connect identifiers**
- Net service names in a `tnsnames.ora` file
- Directory server access

Oracle Managed Files

An automated file system that controls the use of files within Oracle. Oracle Managed Files requires a clustered file system.

Oracle parallel execution

Divides the work of processing certain types of SQL statements among multiple parallel execution server processes. This is not the same as the product formerly known as Oracle Parallel Server (now called Real Application Clusters), which implies multiple **instances** sharing a database.

Oracle Parallel Server

See **Oracle Real Application Clusters**.

Oracle Parallel Server Manager (OPSM)

See **Server Management (SRVM)**.

Oracle Performance Manager

An add-on application for **Oracle Enterprise Manager** that offers a variety of tabular and graphic performance statistics for Real Application Clusters. The statistics represent the aggregate performance for all instances running on a Real Application Clusters database.

Oracle Real Application Clusters

An architecture that allows multiple instances to access a shared database of datafiles. Real Application Clusters is also a software component that provides the necessary Real Application Cluster scripts, initialization files, and datafiles to make the Oracle9*i* Enterprise Edition an Oracle9*i* 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 Oracle9*i* 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 `init`*`db_name`*`.ora` file, and unique **thread ID** make up each node's *`sid`*. The thread ID starts at 1 for the first instance in the cluster, and is incremented by 1 for the next instance, and so on.

For pre-release 8.1 databases, *`sid`* identified the database. The *`sid`* was included in the part of the connect descriptor in a `tnsnames.ora` file, and in the definition of the network listener in the `listener.ora` file.

Oracle Universal Installer (OUI)

A graphical user interface that facilitates the installation of the Oracle database software and its related components.

Performance Manager

An add-on application for Oracle Enterprise Manager that offers a variety of tabular and graphic performance statistics for Real Application Clusters. The statistics represent the aggregate performance for all instances running on Real Application Clusters.

PMON process

A database process that performs process monitoring. PMON also performs recovery when a user process fails. PMON is responsible for cleaning up the cache and freeing resources that the process was using. P MON also checks on **dispatcher** and server processes and restarts them if they have failed. As a part of **service registration**, PMON registers instance information with the listener.

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.

After these credentials are set up, you log in once to start the **Console** and are then automatically logged in as needed to the nodes. All login credentials are encrypted in the **repository database**.

quorum disk

See **voting disk**.

raw device

A raw device is 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

Formerly known as *Oracle Parallel Server*, an architecture that allows multiple instances to access a shared database of datafiles. Real Application Clusters is also a software component that provides the necessary Real Application Clusters scripts, initialization files, and datafiles to make the Oracle9*i* Enterprise Edition an Oracle9*i* Real Application Clusters database.

redo log file

A file that contains a record of all changes made to data in the database buffer cache. If an instance failure occurs, the redo log files are used to recover the modified data that was in memory.

repository database

A repository database is 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)

Recovery Manager (RMAN) is an Oracle tool that backs up, copies, restores, and recovers datafiles, control files, and archived redo logs.

rollback segment

Contains transactions to undo changes to data blocks for uncommitted transactions. Rollback segments also provide read consistency to roll back transactions and to recover the database. Each node typically has two rollback segments that are identified with a naming convention of `RBSthread_id_rollback_number` by the `ROLLBACK_SEGMENTS` parameter in the instance initialization file, **initsid.ora**.

seed database

A preconfigured, ready-to-use database that requires minimal user input to create.

Server Management (SRVM)

A comprehensive and integrated system management solution for Real Application Clusters. Use Server Management to manage multi-instance databases running in heterogeneous environments through an open client/server architecture through **Oracle Enterprise Manager**.

In addition to managing Real Application Clusters, Server Manager enables you to schedule jobs, perform event management, monitor performance, and obtain statistics to tune Real Application Cluster databases.

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.

If you are not sure what the global database name is, you can obtain it from the combined values of the `SERVICE_NAMES` parameter in the common database initialization file, **initdb_name.ora**.

The service name is included in the `CONNECT_DATA` part of the **connect descriptor**.

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.

Service registration provides the listener with the following information:

- Service name(s) for each running instance of the database
- Instance name(s) of the database
- Service handlers (**dispatchers** and dedicated servers) available for each instance

  This allows the listener to direct a client's request appropriately.
- Dispatcher, instance, and node load information

  This load information allows the listener to determine which dispatcher can best handle a client connection's request. If all dispatchers are blocked, the listener can spawn a dedicated server for the connection.

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**. The dispatcher directs multiple incoming network session requests to a common queue. An idle shared server process from a shared pool of server processes picks up a request from the queue. This means a small pool of server processes can server a large amount of clients. 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:

- Client domain to append to unqualified service names or net service names
- Order of naming methods the client should use when resolving a name
- Logging and tracing features to use
- Route of connections
- Preferred Oracle Names servers
- External naming parameters
- Oracle Advanced Security parameters

The `sqlnet.ora` file typically resides in `$ORACLE_HOME/network/admin` on UNIX platforms and `ORACLE_HOME\network\admin` on Windows platforms.

SRVCTL Utility

A utility that manage instances. 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)

Startup is 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.

System Global Area (SGA)

A group of shared memory structures that contain data and control information for an Oracle **instance**.

tablespace

A logical portion of an Oracle database used to allocate storage for table and index data. Each tablespace corresponds to one or more physical datafiles. Every Oracle database has a tablespace called SYSTEM and may have additional tablespaces. A tablespace is used to group related logical structures. For example, tablespaces commonly group all of an application's objects to simplify administrative operations.

thread ID

The number of the redo thread to be used by an instance. Any available redo thread number can be used, but an instance cannot use the same thread number as another instance. Also, an instance cannot start when its redo thread is disabled. An instance cannot mount a database if the thread is used by another instance or if the thread is disabled.

The thread starts at 1 node for the first instance in the cluster, and is incremented by 1 for the next instance, and so on.

Threads are depicted by the THREAD parameter in the instance initialization file, **initsid.ora**.

When redo log files are generated, they include the thread ID, allowing you to easily identify a particular node's log files.

tnsnames.ora

A configuration file that contains **net service name***s* mapped to **connect descriptor***s*. 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 Oracle 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 allows 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)

Transmission Control Protocol/Interconnect Protocol (TCP/IP) is a set of protocols that govern how information is packaged and transferred across the internet to guarantee reliable service.

trace file

Each server and background process can write to an associated trace file. When a process detects an internal error, the process dumps information about the error to its trace file. Some of the information written to the trace file is intended for the database administrator, while other information is intended for Oracle Support Services. Trace file information is also used to tune applications and instances.

User Datagram Protocol (UDP)

A transport layer protocol defined by the US Department of Defence (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 is does not have a delivery guarantee or protection from duplication.

Virtual Interface Architecture (VIA)

Virtual Interface Architecture is

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