Oracle8*i* Parallel Server

Setup and Configuration Guide

Release 2 (8.1.6)

December 1999

Part No. A76934-01

Oracle8i Parallel Server Setup and Configuration Guide, Release 2 (8.1.6)

Part No. A76934-01

Primary Author: Deborah Steiner

Contributors: Mark Bauer, James Cassidy, Kevin Chang, Toby Close, Sashikanth Chandrasekaran, Jonathan Creighton, Harvey Eneman, Hogan Flake, Peter Povinec, and Peter Sciarra

Contents

Part II Installing and Configuring Oracle Parallel Server

2 Pre-Installation

3 Installing and Creating an Oracle Parallel Server Database

4 Additional Configuration Issues

5 Configuring High-Availability Features

Part III Installing and Using Oracle Parallel Server Management

6 Installing and Configuring Oracle Parallel Server Management

7 Administering Oracle Parallel Server with Oracle Parallel Server Management

8 Monitoring Performance with Oracle Performance Manager

Part IV Adding Instances and Nodes

9 Adding Instances and Nodes

Part V Reference

A Directory Structure

B Oracle Parallel Server Management on UNIX Reference

C Troubleshooting

Glossary

Index

List of Figures

Send Us Your Comments

Oracle8i Parallel Server Setup and Configuration Guide, Release 2 (8.1.6)

Part No. A76934-01

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

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

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

- E-mail - infodev@us.oracle.com
- FAX - (650) 506-7228. Attn: Information Development
- Postal service:

  Oracle Corporation
  Server Technologies Documentation Manager
  500 Oracle Parkway, 4OP12
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, and telephone number below.

____

____

____

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

Preface

Oracle8i Parallel Server Setup and Configuration Guide explains how to install, configure, and manage Oracle Parallel Server in an Oracle8*i* environment.

This preface includes the following topics:

- Intended Audience
- Prerequisites
- Related Documents
- Conventions
- Your Comments Are Welcome

See Also:

- *Oracle8i Parallel Server Documentation Online Roadmap* for further information about using Oracle Parallel Server online documentation
- *Oracle8i Parallel Server Concepts* for information on feature changes from one release of Oracle Parallel Server to another

Intended Audience

The information in this guide is intended primarily for network or database administrators (DBAs) responsible installation and configuration of Oracle Parallel Server.

The information in this guide is intended primarily for network administrators and database administrators responsible for the installation and configuration of Oracle Parallel Server.

Prerequisites

This guide assumes you are familiar with:

- The target operating system and have installed and tested it for the cluster and network hardware
- Object relational database management concepts

How this Book is Organized

This guide is divided into five parts:

Part I: Introduction to Installing Oracle Parallel Server	
Chapter 1, "Introduction to the Installation and Configuration Process"	Describes the main features and components of Oracle Parallel Server
Part II: Installing and Configuring Oracle Parallel Server	
Chapter 2, "Pre-Installation"	Describes the specific hardware and software, raw device, and pre-installation requirements for installing Oracle Parallel Server
Chapter 3, "Installing and Creating an Oracle Parallel Server Database"	Describes how to install the components needed for Oracle Parallel Server
Chapter 4, "Additional Configuration Issues"	Describes how to configure clients and the initialization files
Chapter 5, "Configuring High-Availability Features"	Describes how to configure Transparent Application Failover (TAF) and primary and secondary instances
Part III: Installing and Using Oracle Parallel Server Management	
Chapter 6, "Installing and Configuring Oracle Parallel Server Management"	Describes the tasks to install and configure Oracle Enterprise Manager and Oracle Performance Manager for parallel server management
Chapter 7, "Administering Oracle Parallel Server with Oracle Parallel Server Management"	Describes Oracle Parallel Server initialization files and administration of Oracle Parallel Server with the Oracle Enterprise Manager Console
Chapter 8, "Monitoring Performance with Oracle Performance Manager"	Describes how to use Oracle Performance Manager to generate metric charts for Oracle Parallel Server databases
Part IV: Adding Instances and Nodes	
Chapter 9, "Adding Instances and Nodes"	Describes how to add nodes to an existing cluster, and how to migrate from an single-instance database to multiple instances
Part V: Reference	
Appendix A, "Directory Structure"	Describes the directory structure for Oracle Parallel Server
Appendix B, "Oracle Parallel Server Management on UNIX Reference"	Provides reference information for Oracle Parallel Server Management on UNIX
Appendix C, "Troubleshooting"	Describes how to troubleshoot common Oracle Parallel Server configuration issues

Related Documents

For more information, see the following user guides:

Installation Guides

- *Oracle8i Installation Guide* for Sun Solaris, HP 9000 and AIX-based systems
- *Oracle8i Installation Guide for Windows NT*
- *Oracle Diagnostics Pack Installation*

Operating System-Specific Administrative Guides

- *Oracle8i Administrator's Reference* for Sun Solaris, HP 9000 or AIX-based systems
- *Oracle Parallel Server Administrator's Guide for Windows NT*
- *Oracle8i Administrator's Guide for Windows NT*

Oracle Parallel Server Documentation

- *Oracle8i Parallel Server Administration, Deployment, and Performance*
- *Oracle8i Parallel Server Concepts*
- *Oracle8i Parallel Server Documentation Online Roadmap*

Oracle Parallel Server Management

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

Generic Documentation

- *Oracle8i Concepts*
- *Net8 Administrator's Guide*
- *Getting to Know Oracle8i*
- *Oracle8i Reference*

Conventions

The following conventions are used in this book:

Convention	Meaning
UPPERCASE	Calls attention to SQL commands, keywords, and initialization parameters.
bold	Boldface test indicates a term defined in the glossary.
`lowercase courier`	Indicates file names, directories and commands.
`constant width`	This typeface is used for user input and code examples.
Italic	Terms in italics indicate a variable or special emphasis. Italic is also used for book titles.
$ORACLE_HOME on UNIX and *ORACLE_HOME* on Windows platforms	In this Optimal Flexible Architecture (OFA)-compliant release, all subdirectories are no longer under a top level `ORACLE_HOME` directory. There can be an `ORACLE_BASE` directory, which may contain multiple Oracle home directories.
`< >`	Angle brackets enclose user-supplied names.
`[ ]`	Brackets enclose a choice of optional items from which you can choose one or none.
`{ }`	Curly brackets enclose required items.
`...`	Horizontal ellipsis points in code samples mean that parts of the sample have been omitted.

Your Comments Are Welcome

We value and appreciate your comment as an Oracle user and reader of our manuals. As we write, revise, and evaluate our documentation, your opinions are the most important feedback we receive.

You can send comments and suggestions about this reference to the Information Development department at the following e-mail address:

infodev@us.oracle.com

If you prefer, you can send letters or faxes containing your comments to:

Server Technologies Documentation Manager
Oracle Corporation
500 Oracle Parkway, 4OP12
Redwood Shores, CA 94065
Fax: (650) 506-7228 Attn.: Information Development

Part I

Introduction to Installing Oracle Parallel Server

Part I provides an overview of the components required to install and configure the Oracle Parallel Server Option. The chapter in this part is:

- Chapter 1, "Introduction to the Installation and Configuration Process"

1

Introduction to the Installation and Configuration Process

This chapter provides an overview of the Oracle Parallel Server installation process by describing the components required for configuring and deploying Oracle Parallel Server. This information also helps you prepare and plan your Oracle Parallel Server installation and configuration process in an Oracle8*i* environment.

This chapter includes the following topics:

- Introduction to Oracle Parallel Server
- Oracle Parallel Server Software Components
- Installation Overview
- Database Configuration Overview
- Selecting a Database Creation Method

Introduction to Oracle Parallel Server

Oracle Parallel Server is an architecture that enables multiple instances to access a shared database. Oracle Parallel Server offers the following:

- Superior scalability through distribution of workload across nodes
- High-availability through multiple nodes accessing the database; if one node fails, the database is still accessible through surviving nodes

Each Oracle Parallel Server **instance**, a running instance of Oracle8*i* software, is comprised of a **System Global Area (SGA)** and Oracle background processes. The SGA is a shared memory region that contains data and control information for an instance. Users can connect to any instance to access the information that resides within the shared database.

> **See Also:** *Oracle8i Parallel Server Concepts*

Oracle Parallel Server instances coordinate with the following components:

Component	Description
Node	A server where an instance resides
Cluster	A set of physically interconnected nodes and a shared disk storage subsystem
Database	The set of all instances with access to the same data files. A database is limited to a set of Oracle Parallel Server instances that run on only the nodes defined within a cluster. The database name is defined by *global database name*, a name comprised of the database name and domain, entered during installation. The database name is unique in the cluster.

All instances share:

- The same set of **data files**
- The same **control files** to start and run the database

The data files and control files are located on raw disk devices that are shared between multiple nodes.

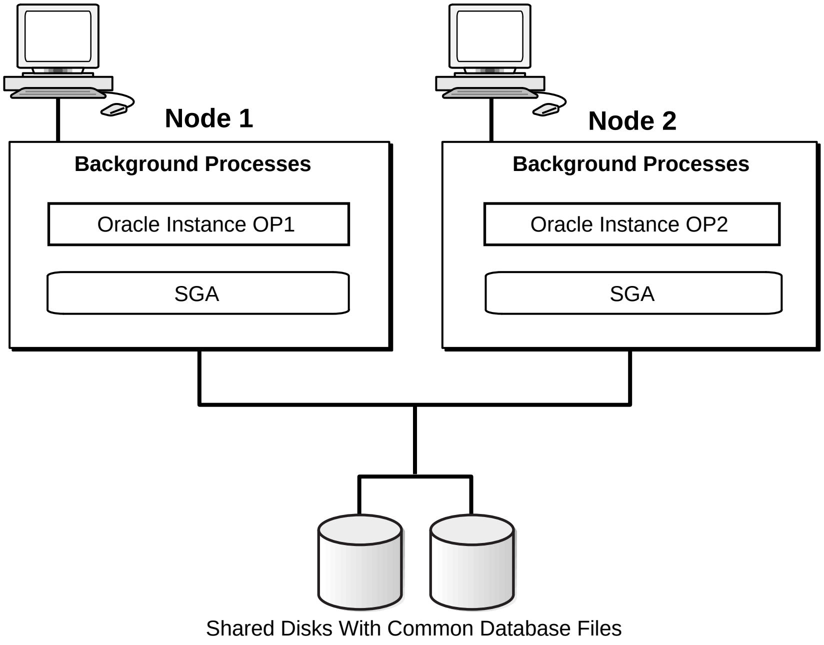
If one node fails, client applications (written to do so) can re-route users to another node. One of the surviving nodes automatically performs recovery by rolling back any incomplete transactions that the other node was attempting. This ensures the logical consistency of the database.

Note: Because an instance does not include data files, you can start up an instance without mounting the data files.

Each instance has its own set of **redo log files**. Although instances use their own redo log files, the redo log fields must exist on the raw devices. This way, each node can perform recovery on behalf of another node.

Figure 1–1 shows the relationship between two Oracle instances and the shared disks on which the data files are stored:

Figure 1–1 Oracle Parallel Server Instance and Shared Data File Architecture



Oracle Parallel Server Software Components

The following components comprise Oracle Parallel Server:

Component	Description
Oracle8*i* Enterprise Edition	Provides the applications and files to manage a database. All other Oracle Parallel Server components are layered on top of Oracle8*i* Enterprise Edition
Oracle Parallel Server	Provides the necessary Oracle Parallel Server scripts, initialization parameter files, and data files
Oracle Parallel Server Management	Provides a single point for starting, stopping, and monitoring the activity of parallel servers and parallel server instances from within Oracle Enterprise Manager Console **See Also:** ■ Chapter 5 ■ Chapter 6
Operating System Dependent layer	Consists of several software components developed by vendors. The Operating System Dependent layer maps the key OS/cluster-ware services required for proper operation of Oracle Parallel Server. **See Also:** "Operating System Dependent Layer" below

Operating System Dependent Layer

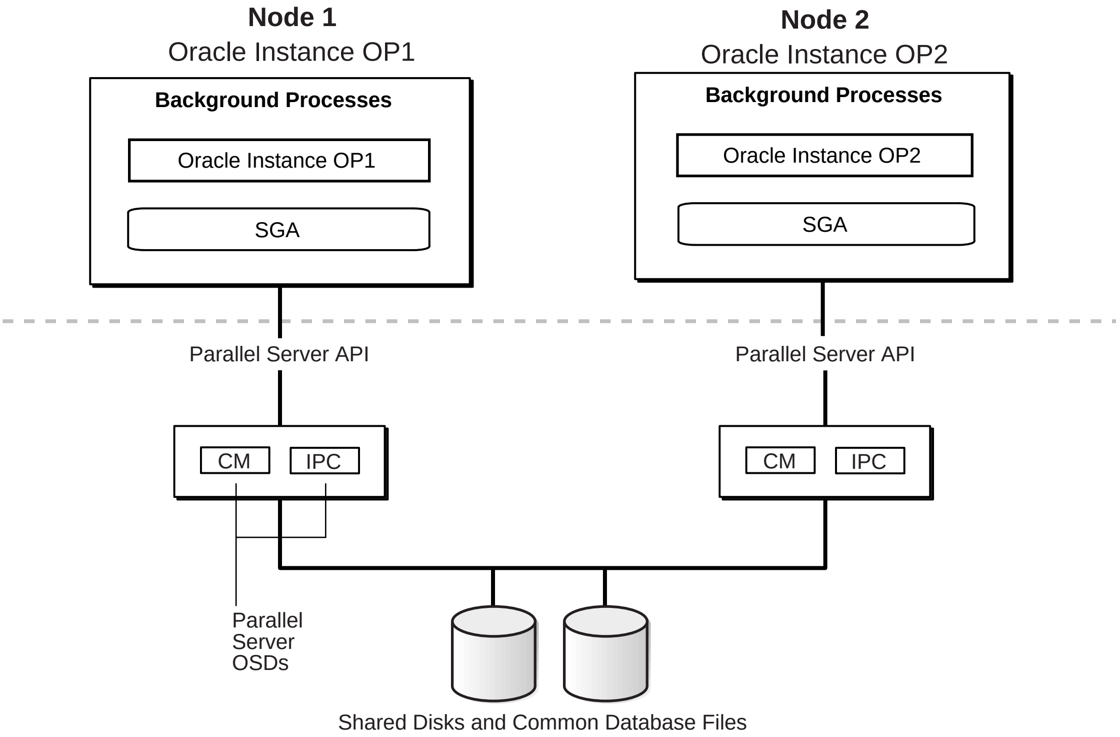
You must install a vendor-supplied, Oracle-certified Operating System Dependent layer before you install Oracle Parallel Server software. The Operating System Dependent layer maps the key OS/cluster-ware services required for Oracle Parallel Server. This layer consists of two primary software components developed by vendors, as described in the following table:

Component	Description
Cluster Manager (CM)	Discovers the state of the cluster and instance membership
Inter-Process Communication (IPC)	Provides reliable transfer of messages between instances on different nodes

These components provide key services required for proper operation of Oracle Parallel Server and are used by various clients, such as Integrated Distributed Lock Manager.

Figure 1–2 illustrates the Operating System Dependent components in a cluster with two nodes:

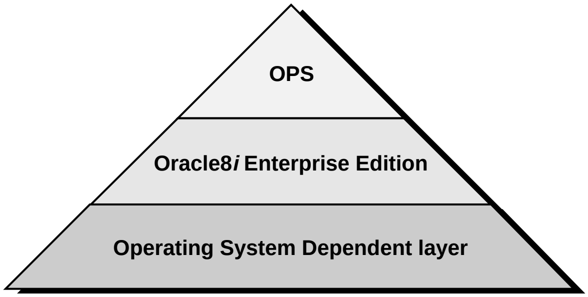
Figure 1–2 Operating System Dependent Components



Installation Overview

Oracle Parallel Server software installation requires the following components:

- Operating System Dependent layer
- Oracle8*i* Enterprise Edition
- Oracle Parallel Server


OPS
Oracle8*i* Enterprise Edition
Operating System Dependent layer

Similar to building blocks, the components must be installed in the following order:

1. Install the Operating System Dependent layer, as prescribed by your vendor.
2. Install Oracle8*i* Enterprise Edition and Oracle Parallel Server from the Oracle8*i* Enterprise Edition CD-ROM.

> **Note:** If you intend to build an Oracle Parallel Server database during installation, follow the raw device creation guidelines in Chapter 2. Raw device creation is necessary because **Oracle Database Configuration Assistant,** which is launched during installation, expects the devices to be established.

Database Configuration Overview

Once Oracle Parallel Server components are installed, the database can be created. Database creation involves two steps:

1. Establish raw devices for:
   - Shared data files
   - Shared control files
   - Each node's redo log files
2. Create the database using either Oracle Database Configuration Assistant application or manual methods.

Selecting a Database Creation Method

The database can be created using Oracle Database Configuration Assistant or using manual methods. The table below describes the two methods:

Creation Method	Description
Oracle Database Configuration Assistant	Oracle Database Configuration Assistant is a graphical user interface (GUI) tool that enables you to create a database for an Online Transaction Processing (OLTP), Data Warehousing, or Multi-Purpose environment. Oracle Database Configuration Assistance creates a database using the **Optimal Flexible Architecture (OFA)**, whereby database files and administrative files, including initialization parameter files, follow standard naming and placement practices. Additionally, it creates a standard set of tablespaces. Oracle Database Configuration Assistant can be automatically started from within Oracle Universal Installer when you select to create a database or manually run as a stand-alone tool.
Manual	The database should be created manually if you already have existing scripts or require different data files and tablespaces than those created by Oracle Database Configuration Assistant.

Using Oracle Database Configuration Assistant

This section contains the following sections:

- Advantages of Using Oracle Database Configuration Assistant
- Identifying Your Database Environment
- Selecting a Database Creation Method
- Looking at a Database Created by Oracle Database Configuration Assistant

Advantages of Using Oracle Database Configuration Assistant

Here are some of the advantages of using Oracle Database Configuration Assistant:

- It uses Optimal Flexible Architecture (OFA).
- It is fast. A ready made database can be copied into place rather than going through a lengthy creation process. This database is a Multi-Purpose database, suiting many users' environments. It can be customized later, if desired.
- You do not need to spend time deciding how to set parameters. Decisions have already been made for you.
- It customizes your database for you. Even if you do not choose to copy the starter database, you can still direct Oracle Database Configuration Assistant to generate a script that creates an online transaction processing (OLTP), Data Warehousing, or Multi-Purpose database. You need answer only a few questions presented to you by Oracle Database Configuration Assistant. It automatically includes **multi-threaded server (MTS)** parameters if you have over a specified number of users.

Identifying Your Database Environment

Oracle Database Configuration Assistant enables you to create a database in one of the following environments. Identify the environment appropriate for your database:

Environment	Description
Online Transaction Processing (OLTP)	Databases in OLTP environments must process thousands or even millions of transactions from many concurrent (simultaneously connected) users each day. These transactions consist of reading (SELECT statements), writing (INSERT and UPDATE statements), and deleting (DELETE statements) data in database tables. Users must have quick access to the most current data. Therefore, database performance is defined in terms of throughput (speed) and availability of data.
Data Warehousing	Databases in Data Warehousing environments must process a variety of queries (typically read-only), ranging from a simple fetch of a few records to numerous complex queries that sort thousands of records from many different tables. Therefore, database performance is defined in terms of response time.
Multi-Purpose	Multi-Purpose databases support both OLTP and Data Warehousing environments.

Selecting a Database Creation Method

When you run Oracle Universal Installer and select Oracle8*i* Enterprise Edition, you are presented with choice of Typical or Custom installation. Each installation type enables you to create a database.

The types of database (OLTP, Data Warehousing, and Multi-Purpose) created with the Typical and Custom installation types and the amount of user input are described below. Review these selections to identify the database that best matches your database requirements and database creation expertise.

If You Perform These Steps...	Then...
1. Select the Typical installation type.	Oracle Database Configuration Assistant automatically starts at the end of installation and copies a preconfigured, ready-to-use Multi-Purpose database from the CD-ROM. This is referred to as a **seed database** or **starter database**. No user input is requested, making creation efficient.
1. Select the Custom installation type. **2.** Select Yes when prompted to create a database **3.** Oracle Database Configuration Assistant prompts you to select either of two choices: - Typical database creation type - Custom database creation type	Oracle Database Configuration Assistant role in database creation depends upon your selection: **Typical database creation type** You have two choices. Oracle Database Configuration Assistant role in database creation depends upon your selection. If you select: ■ **Copy existing database files from the CD** Oracle Database Configuration Assistant creates the same database as during Typical installation. No user input is required. ■ **Create new database files** Asks you several database environment questions before dynamically creating a database script for an OLTP database, Data Warehousing database, or Multi-Purpose database with standard init*sid*.ora and init*db_name*.ora files. **Note:** If 20 or more users are specified for an OLTP database, MTS parameters are set in the init*db_name*.ora file. **Custom database creation type** Oracle Database Configuration Assistant guides you in the creation of database fully customized to match the environment (OLTP, Data Warehousing, Multi-Purpose) and database configuration mode (dedicated server or multi-threaded server) you select. Select this option only if you are experienced with advanced database creation procedures, such as customizing: ■ MTS parameters ■ Data, control, and redo log file settings ■ Tablespace sizes ■ Database memory parameters ■ Archiving formats and destinations ■ Trace file destinations ■ Character set value

Looking at a Database Created by Oracle Database Configuration Assistant

The database configured with Oracle Database Configuration Assistant contains the following features:

- Tablespaces and Data Files
- Redo Log Files
- Control Files
- Rollback Segments
- Initialization Parameter Files
- Unique Identification of Instances with the initsid.ora File

Tablespaces and Data Files

An Oracle Parallel Server database is divided into smaller logical areas of space known as **tablespaces** Each tablespace corresponds to one **data file**, stored on a raw device. The tablespace names used in an Oracle Parallel Server database contain the following types of data:

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 Oracle8*i* databases.
USER	Consists of application data. As you create and enter data into tables, you fill this space with your data.
TEMP	Temporary tables and/or indexes created during the processing of your SQL statement. You may need to expand this tablespace if you are executing a SQL statement that involves a lot of sorting, such as ANALYZE COMPUTE STATISTICS on a very large table, or the constructs GROUP BY, ORDER BY, or DISTINCT.
RBS	Consists of the rollback segments. Rollback segments are sets of extents that record data before it is modified by transactions, enabling the changes to be undone.
INDX	Stores indexes associated with the data in the USER tablespace
TOOLS	Stores tables for Oracle Enterprise Manager
DRSYS	Consists of data for Oracle8*i* *inter*Media Text

The tablespace names cannot be altered with the Typical database creation type. You can change the names of the tablespaces, except for SYSTEM, with the Custom database creation type.

Each tablespace is represented by one database file. The data file names created by the Typical database creation type vary by operating system. UNIX prompts you to set the file names. Windows NT uses the **symbolic link names** shown in the following table:

Tablespace	Windows NT Symbolic Link
SYSTEM	`\\.\`*db_name*`_system1`
USERS	`\\.\`*db_name*`_users1`
TEMP	`\\.\`*db_name*`_temp1`
RBS	`\\.\`*db_name*`_rbs1`
INDX	`\\.\`*db_name*`_indx1`
TOOLS	`\\.\`*db_name*`_tools1`
DRSYS	`\\.\`*db_name*`_drsys`

You can specify different symbolic names with the Custom database creation type

Redo Log Files

Each node is configured with two **redo log files**. Redo log files are used to hold a record of all changes made to data. If a database failure occurs before the data is written to disk, the changes can be obtained from the redo log files. This prevents data from ever being lost. Redo log files are stored on the raw devices.

Redo log files are used in a cyclical fashion. The first file is filled first, then the second file. The first file is then re-used and filled, the second file is re-used and filled, and so on.

The redo log files created with the Typical database creation type vary by operating system. UNIX prompts you to set the file names. Windows NT uses symbolic link names of `\\.\`*db_namethread_number*, where *thread* is the thread ID of the node, and *number* is the number (1 or 2) of the redo log file. The Custom database creation type prompts you to specify redo log file names or symbolic link names.

Control Files

The database is configured with two **control files**. A control file is an administrative file required to start and run the database. The control file records the physical structure of the database. For example, a control file contains the database name, and the names and locations of the database's data files and redo log files. The control files are stored on the raw devices.

The controls files created by the Typical database creation type vary by operating system. UNIX prompts you to set the file names. Windows NT uses symbolic link names of `\\.\control1` and `\\.\control2`. The Custom database creation type prompts you to specify control file names or symbolic link names.

Rollback Segments

Each node is configured with two **rollback segments**. Rollback segments record the old values of data changed by each transaction (whether or not committed). Every database contains one or more rollback segments, which are portions of the database that record the actions of transactions in the event that a transaction is rolled back. Rollback segments are used to provide read consistency, to roll back transactions, and to recover the database. Note that rollback segments are database objects, not files that reside in the RBS tablespace.

The rollback segments created by the Typical or Custom database creation type have names of `rbs`*`thread_number`*, where *thread* is the thread ID of the node, and *number* is the number (1 or 2) of the rollback segment.

Initialization Parameter Files

Each node has an instance-specific initialization parameter file and an Oracle Parallel Server database initialization parameter file.

The Typical database creation type stores the initialization parameter file in `$ORACLE_BASE/admin/db_name/pfile` on UNIX and *`ORACLE_BASE`*`\admin\`*`db_name`*`\pfile` on Windows NT. The table below describes the initialization parameter files:

Initialization Parameter File	Description
`init`*`sid`*`.ora`	Each node's instance has its own `init`*`sid`*`.ora` file, where *sid* is the **Oracle System Identifier (SID)** of the instance. This file uniquely defines the instance with instance-related parameters. This file calls the`init`*`db_name`*`.ora` file.

Initialization Parameter File	Description
init*db_name*.ora	Parameters for an Oracle Parallel Server database are stored in init*db_name*.ora file, where *db_name* is the database name. It lists the common database parameters shared from node-to-node. This file resides on each of the nodes.

The Custom database type allows you to set the location of the init*db_name*.ora file.

> **See Also:** "Understanding the Initialization Parameter Files" on page 4-4 for further information about these files.

Unique Identification of Instances with the init*sid*.ora File

The init*sid*.ora file is influenced by a number of elements, including:

Element	Description
Oracle System Identifier (SID)	Uniquely identifies a node's instance 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. The SID is defined as an ORACLE_SID environment variable on UNIX platforms and in the ORACLE_SID registry entry on Windows NT.
Instance Name	Represents the name of the instance and is used to uniquely identify a specific instance when multiple instances share common service names. The instance name is the same as the SID. It is indicated by the INSTANCE_NAME parameter in the init*sid*.ora file.

Element	Description
Instance Number	Associates extents of data blocks with particular instances The instance number ensure that an instance uses the extents allocated to it for inserts and updates. 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 indicated by the INSTANCE_NUMBER parameter in the `init`*`sid`*`.ora` file.
Rollback Segments	Each node is configured with two rollback segments. Rollback segments record the old values of data changed by each transaction (whether or not committed). Every database contains one or more rollback segments, which are portions of the database that record the actions of transactions in the event that a transaction is rolled back. Rollback segments are used to provide read consistency, to roll back transactions, and to recover the database. Oracle Database Configuration Assistant creates rollback segment identified by rbs*thread_number* where *thread* is the thread ID of the node, and *number* is the number (1 or 2) of the rollback segment. For example, the first node in a cluster would have rollback segments of rbs1_1 and rbs1_2. Rollback segments are indicated by the ROLLBACK_ SEGMENTS parameter in the `init`*`sid`*`.ora` file.
Thread ID	Specifies the redo thread that is to be used by the 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. Redo thread numbers are used in the naming of redo log files, which record changes made to data. When redo log files are generated, they include the thread, allowing you to easily identify a particular node's redo log files. Threads are indicated by the THREAD parameter in the `init`*`sid`*`.ora` file.

If the database name is op and the thread IDs are 1, 2 and 3, then the SID for each node is:

Thread ID	SID
1	op1
2	op2
3	op3

If the DB_NAME is op and the thread IDs are 1, 2 and 3, then the instance elements are affected in the following manner:

Computer	Thread ID	SID	INSTANCE_NAME	ROLLBACK_SEGMENT name
Computer A	1	op1	op1	rbs1-1, rbs1-2
Computer B	2	op2	op2	rbs2-1, rbs2-2
Computer C	3	op3	op3	rbs3_1, rbs3_2

Using Manual Methods

Oracle Database Configuration is intended primarily for new installations and configurations. If you already have existing scripts that are customized to your environment, it is advisable to create the database manually after installation, as described in "Use Manual Methods" on page 3-29.

Part II

Installing and Configuring Oracle Parallel Server

Part II describes the Oracle Parallel Server installation and configuration process. The chapters in Part II are:

- Chapter 2, "Pre-Installation"
- Chapter 3, "Installing and Creating an Oracle Parallel Server Database"
- Chapter 4, "Additional Configuration Issues"
- Chapter 5, "Configuring High-Availability Features"

2

Pre-Installation

This chapter describes pre-installation requirements for Oracle Parallel Server.

Specific topics discussed are:

- System Installation Requirements
- Setting Up Raw Devices
- Pre-Installation Steps

System Installation Requirements

Verify that your system meets the installation requirements described in the following sections before you install.

- Hardware and Software Requirements for Oracle8i Parallel Server Nodes
- Hardware and Software Requirements for Oracle Enterprise Manager
- Shared Disk Subsystem

Hardware and Software Requirements for Oracle8*i* Parallel Server Nodes

Verify the hardware 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

Software

Each node in a cluster requires the following software:

- ❑ Operating-system specific software, as described in operating-system-specific installation guides.
- ❑ Operating System Dependent layer from a vendor that has passed certification
- ❑ Oracle8*i* Enterprise Edition
- ❑ Net8 Server
- ❑ Oracle Parallel Server
- ❑ Oracle Intelligent Agent release 8.1.6 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`

Hardware and Software Requirements for Oracle Enterprise Manager

Oracle Enterprise Manager version 2 is a management framework consisting of a Console, a suite of tools and services, and a network of management servers and Oracle Intelligent Agents.

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.

The components are listed below:

- Console and DBA Studio
- Management Server
- Repository database
- Managed targets with agents throughout the network

Component	Description
Console and DBA Studio	The **Console** works with **Oracle Intelligent Agents** (installed on the nodes) to perform database administration from a single console. If DBA Studio is on its own machine or is run in stand-alone mode without a connection to the Management Server, configuration of a `tnsnames.ora` file is required. This file should contain entries for instances.
Management Server	The **Management Server** performs functions requested by the Console. If the Management Server is on its own machine separate from the repository database, Net8 installation and configuration is required to communicate with the repository database.
Repository database	A **repository database** is a set of tables in an Oracle database which stores data required by Oracle Enterprise Manager. You must store the repository in an Oracle database version 7.3.4 or above.
Managed targets with agents throughout the network	A managed target is any service or entity which can be centrally managed by Oracle Enterprise Manager. Examples are nodes, databases, web servers, listeners, and paging services. Each Oracle Parallel Server must have a release 8.1.6 or earlier Oracle Intelligent Agent installed.

Operating System

- Windows NT version 4.0
- Solaris 2.6

Note: The Console and DBA Studio may also run on Windows 95 or Windows 98. The Management Server and repository cannot run on these operating systems.

Enterprise Manager Software

- Oracle Enterprise Manager version 2
- Oracle Diagnostics Pack version 2

  Oracle Diagnostics Pack includes Oracle Performance Manager application, from which you can choose from a variety of graphic statistical performance charts for parallel servers.

Note: You may also use prior releases of Oracle Enterprise Manager and Oracle Diagnostics Pack for a release 8.1.6 database. See Oracle Enterprise Manager documentation for compatibility information.

- Net8 Client on the machine where the Diagnostics Pack is installed. Net8 Client is also required if the Management Server and Database Administration Applications and Console are on their own machines.

Supported Oracle Database Versions as Repositories

8.1.6, 8.1.5, 8.0.5, 8.0.4, 8.0.3, 7.3.4 on database repository machine

Supported Oracle Intelligent Agents

8.1.6 on all Oracle Parallel Server nodes

Oracle Intelligent Agent may be installed from the Oracle8*i* Enterprise Edition CD-ROM.

See Also: Oracle installation guide for detailed disk space and RAM requirements

Shared Disk Subsystem

Oracle Parallel Server requires a shared disk subsystem to contain shared partitions that are raw. All Oracle8*i* data, log, and control files are placed on shared raw partitions.

Note: Each instance of an Oracle Parallel Server database has its own log files, but control files and data files are shared by instances in the cluster. However, log files must be accessible/readable by other instances.

Setting Up Raw Devices

Clusters do not provide access to a shared file system among all nodes of a cluster. As a result, data files, redo log files, and control files are stored on raw devices. All instances share the data files and control files. However, each instance has its own redo log files, but all instances must have access to all log files during recovery.

Make sure you set up raw devices prior to installation If you run **Oracle Database Configuration Assistant** without setting up raw devices, the database cannot be created.

In order for Oracle Database Configuration Assistant to create the files for the database, a precise number of raw devices must be set up prior to database creation. These raw devices include:

- Six for data files
- One for Oracle8*i* *inter*Media (if used)
- Two for control files
- Two for redo log files for each node

The Typical database creation type available with Oracle Database Configuration Assistant creates the files at the sizes indicated in the table below. Raw partitions should be at least 1 MB larger than the file sizes.

Raw Device Must be Created For	File Size
SYSTEM tablespace	200 MB
USERS tablespace	108 MB
TEMP tablespace	72 MB for a Multi-Purpose or Online Transaction Processing (OLTP) database 520 MB for a Data Warehousing database **Note:** Multi-Purpose is the default for the Typical installation type.
RBS tablespace	520 MB for a Multi-Purpose or OLTP database 1032 MB for a Data Warehousing database
INDX tablespace	58 MB
TOOLS tablespace	12 MB
DRYSYS tablespace	80 MB
First control file	100 MB
Second control file	100 MB
Two redo log files for each node	1 MB

The Custom database creation type available with Oracle Database Configuration Assistant enables you to specify the file and the block size. Ensure that the raw partitions provide enough space to account for the customized sizes.

If you do not plan to create the database with Oracle Database Configuration Assistant, the number of raw devices created depends on the number of the data files, redo log files and control file you plan to create.

See Also: "Selecting a Database Creation Method" on page 1-8 for more information about databases creation

Note: You can modify file sizes later, but raw devices on Windows NT do not expand or shrink once created. Therefore, file sizes cannot expand the size of the raw device.

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

- UNIX
- Windows NT

UNIX

Use the following procedure to create raw devices on UNIX operating systems:

1. Create raw devices.

   See Also: *Oracle8i Administrator's Reference* for your UNIX operating system

   Note: You must have the `root` privilege to create the raw devices.

2. Create the data files, control files, and redo log files. You can use any file names. For simplicity, Oracle recommends using file names that match the raw device type, for example:

Example File Name	Raw Device
*db_name*_`system01.dbf`	SYSTEM tablespace raw device
*db_name*_`user01.dbf`	USERS tablespace raw device
*db_name*_`temp01.dbf`	TEMP tablespace raw device
*db_name*_`rbs01.dbf`	RBS tablespace raw device
*db_name*_`indx01.dbf`	INDX tablespace raw device
*db_name*_`tools01.dbf`	TOOLS tablespace raw device
*db_name*_`drysys01.dbf`	DRYSYS raw device
*db_name*_`control01.clt`	First control file raw device
*db_name*_`control02.clt`	Second control file raw device
*db_name*_`redo`*thread_number*`.log` *thread* is the thread ID of the node and *number* is the log number (1 or 2) of the node.	Redo log files for each node

3. If you plan to use the Typical installation type, perform the following steps:

   Note: This step is not required for the Custom installation type.

   a. On the node from which you intend to run Oracle Universal Installer, create an ASCII file with entries for each raw device file name, using the format:

   ```
   database_object raw_device_file
   ```

   where *database_object* represents raw device object, and *raw_device_file* is the path of the data file, control file or redo log file. Oracle Database Configuration Assistant expects the database objects listed in the table below:

Database Object...	Used For...
`system1`	SYSTEM tablespace data file
`users1`	USERS tablespace data file
`temp1`	TEMP tablespace data file
`rbs1`	RBS tablespace data file
`indx1`	INDX tablespace data file
`tools1`	TOOLS tablespace data file
`drysys1`	DRYSYS tablespace data file
`control1`	First control file
`control2`	Second control file
redo*thread_number* *thread* is the thread ID of the node and *number* is the log number (1 or 2) of the node.	Redo log files Two entries for each node required. Entries for the first node would look like: redo1_1 redo1_2 Entries for the second node would look like: redo2_1 redo2_2

   The ASCII file should look like the following example for a two-node cluster:

```
system1         device/path/op_system1.dbf
users1          device/path/op_user1.dbf
temp1           device/path/op_temp1.dbf
rbs1            device/path/op_rbs1.dbf
indx1           device/path/op_indx1.dbf
tools1          device/path/op_tools1.dbf
drsys1          device/path/op_drsys1.dbf
control1        device/path/op_control1.clt
control2        device/path/op_control2.clt
redo1_1         device/path/op_redo1_1.log
redo1_2         device/path/op_redo1_2.log
redo2_1         device/path/op_redo2_1.log
redo2_2         device/path/op_redo2_2.log
```

b. Set the environment variable DBCA_RAW_CONFIG so it points to the location of this ASCII file.

When Oracle Database Configuration Assistant creates the database, it looks for the environment variable, reads in the ASCII file, and uses the file names indicated in the right-hand column of the table as the data files when building the tablespaces.

Windows NT

Windows NT does not support a true distributed file system. Therefore, data files, control files and redo log files reside on unformatted raw devices.

An extended partition points to raw space on the disk that can be assigned multiple logical partitions for the database files.

Because raw devices on Windows NT do not have a file name or drive letter associated with them like a regular file system, an **extended partition** is first created, which points to raw space on a disk. Multiple logical partitions are then created and assigned **symbolic link names** using the following format:

```
\\.\symbolic_link_name.
```

A symbolic link is simply a name for a logical partition, such as `\\.\op_system1` for the SYSTEM tablespace. When the SYSTEM tablespace is created, a copy of the data file is made to `\\.\op_system1`, which links to a specific logical partition.

When Oracle Database Configuration Assistant creates the database, it verifies that the symbolic links names have been created and stores the files on the raw devices.

On Windows NT, create logical partitions and symbolic links for the database you are creating.

> **See Also:** *Oracle Parallel Server Administrator's Guide for Windows NT* for further information about creating logical partitions and symbolic links

The Typical database creation type performed by Oracle Database Configuration Assistant requires the following symbolic link names listed in the following table:

Symbolic Link Name...	**Used for...**
`\\.\`*`db_name`*`_system1`	SYSTEM tablespace data file
`\\.\`*`db_name`*`_users1`	USERS tablespace data file
`\\.\`*`db_name`*`_temp1`	TEMP tablespace data file
`\\.\`*`db_name`*`_rbs1`	RBS tablespace data file
`\\.\`*`db_name`*`_indx1`	INDX tablespace data file
`\\.\`*`db_name`*`_tools1`	TOOLS tablespace data file
`\\.\`*`db_name`*`_drsys1`	DRSYS tablespace data file
`\\.\`*`db_name`*`_control1`	First control file
`\\.\`*`db_name`*`_control2`	Second control file
`\\.\`*`db_name`*`_redo`*`thread_ number`* *thread* is the thread ID of the node and *number* is the log number (1 or 2) for the node.	redo log files Each node must have two redo log files. If the database name is OP, link names for the first node would look like: op_redo1_1 op_redo1_2 Link names for the first node would look like: op_redo2_1 op_redo2_2

Pre-Installation Steps

Perform the following steps prior to installation:

1. Install vendor-supplied Operating System Dependent layer. This layer must be Oracle certified.

2. Perform diagnostics on the clusterware as described in your vendor documentation.

3. Create raw devices, as described in "Setting Up Raw Devices" on page 2-5.

4. For UNIX clusters, perform the following as the `root` user:

   a. Make sure you have an OSDBA group defined in the `/etc/group` file on all nodes of the cluster. The OSDBA group name and number (and OSOPER group if you plan to designate one during installation) must be identical for all nodes of a UNIX cluster accessing a single database. The default UNIX group name for the OSDBA and OSOPER groups is `dba`.

   b. Create an *oracle* account on each node of the cluster so that:

      - the account is a member of the OSDBA group

      - the account is used only to install and update Oracle software

      - the account has write permissions on remote directories

   c. Create a mount point directory on each node to serve as the top of your Oracle software directory structure so that:

      -the name of the mount point on each node is identical to that on the initial node

      -the *oracle* account has read, write, and execute privileges

   d. On the node from which you plan to run Oracle Universal Installer, set up user equivalence by adding entries for all nodes, including the local node, in the cluster to the `.rhosts` file of the *oracle* account, or the `/etc/hosts.equiv` file.

   e. Exit the `root` account when you are done.

5. For UNIX clusters, as the *oracle* account, check for user equivalence for the `oracle` account by performing a remote login (`rlogin`) to each node in the cluster. If you are prompted for a password, the *oracle* account has not been given the same attributes on all nodes. Oracle Universal Installer cannot use the `rcp` command to copy Oracle products to the remote directories without user equivalence.

Note: UNIX clusters also require environment setup similar to a single-instance environment. For these instructions and other operating system-specific Oracle Parallel Server pre-installation instructions, see the *Oracle8i Installation Guide* for your UNIX operating system

3

Installing and Creating an Oracle Parallel Server Database

This chapter describes how to install Oracle Parallel Server software and how to create an Oracle Parallel Server database.

Specific topics discussed are:

- Installation Types
- Installation Procedure
- Typical Installation
- Custom Installation
- Understanding the Installed Configuration
- Migrating or Upgrading to Release 8.1
- Multiple Oracle Homes
- Creating the Database After Installation
- Starting the Database in Parallel Mode
- Verifying Instances Are Running
- Deleting the Database

Installation Types

The type of installation and the type of database you choose to install determines how you proceed with installation and configuration decisions.

When you run Oracle Universal Installer, you can choose to install Oracle8*i* Enterprise Edition and Oracle Parallel Server with the following installation types:

Installation Type	Description
Typical	Installs a preconfigured starter database, licensable Oracle options (including Oracle Parallel Server), networking services, Oracle8*i* utilities, and online documentation. This type of installation is recommended for new users or experienced users who want the complete database package. At the end of installation, Oracle Database Configuration Assistant runs to create an Oracle Parallel Server database and initialize the database.
Custom	Enables you to selectively install any product from the CD-ROM, including Oracle8*i* Enterprise Edition and Oracle Parallel Server on any subset of nodes in the cluster. If Oracle8*i* Enterprise Edition is chosen, you can choose to run Oracle Database Configuration Assistant. From Oracle Database Configuration Assistant, you may create a: ■ Starter database ■ Online Transaction Processing (OLTP) database ■ Decision Support System (DSS) database ■ Multi-Purpose database; a mixture of OLTP and DSS processing If you do not run Oracle Database Configuration Assistant, you can run it later to configure Oracle Parallel Server. If you do not install a database, you can manually create the database.

See Also: "Using Oracle Database Configuration Assistant" on page 1-9 for further information about database creation with Oracle Database Configuration Assistant

Note: The Minimal install type does not support Oracle Parallel Server.

Installation Procedure

During installation, software components are installed on the node from which Oracle Universal Installer is run and pushed to the selected nodes in the cluster.

To install Oracle8*i* Enterprise Edition and Oracle Parallel Server:

1. Ensure the tasks in "Pre-Installation Steps" on page 2-11 were performed.
2. Before running Oracle Universal Installer, from the node where you intend to run the installer, verify you have administrative privileges to the other nodes using one of the following operating-system-specific procedures:

Operating System	Verification Method
On UNIX	Perform a remote copy (`rcp`) to the other nodes, including the local node.

On Windows NT	Enter the following command for each node that is a part of the cluster:
	`NET USE \\`*`host_name`*`\C$`
	where *host_name* is the host name defined in the DefinedNodes registry value for Cluster Manager.
	A successful connection results in the following message:
	`The command completed successfully.`
	Oracle 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 username and password which has administrative privileges on each node.

3. Install Oracle8*i* Enterprise Edition on one node in the cluster.

See Also:

- *Oracle8i Installation Guide* for your UNIX operating system
- *Oracle8i Installation Guide for Windows NT*

4. Take note of the following tasks to perform a successful Oracle Parallel Server installation:

   a. Select Oracle8*i* Enterpise Edition from the Available Products page.

   b. Select either Typical or Custom from the Installation Types page. For further information about these install types, see "Typical Installation" on page 3-5 or "Custom Installation" on page 3-5.

   c. If performing a Custom installation, ensure Oracle Parallel Server under the Oracle Database Options folder is selected in the Available Product Components page:

      Oracle Universal Installer does not make Oracle Parallel Server visible unless Cluster Manager clusterware was properly configured. See your OSD vendor documentation for further information.

   d. Select the nodes to which you want the software installed from the Cluster Node selection screen page.

      The node on which you are running the installation is selected whether or not you select it.

      If the list of nodes does not include nodes you expect, the vendor-supplied clusterware is either not installed, not running, or not properly configured.

      See Also: Vendor documentation

      If clusterware is not installed, click Previous, install **Cluster Manager (CM)** on these nodes, then click Next. The nodes should now be listed.

   e. In the Database Identification page, enter an appropriate **global database name**, a name comprised of the database name and database domain, such as `op.us.acme.com`, and accept or change the common prefix that is to be used for the **Oracle System Identifier (SID)** for each instance.

      Each instance has a SID that is comprised of the common prefix entered here and a thread ID that is automatically generated. For example, if `op` is entered, the first instance in the cluster is given a SID of `op1`, and the second instance is given a SID of `op2`.

      During installation no message appears to indicate components are installed to the other nodes.

Typical Installation

If you chose a Typical installation type, **Net8 Configuration Assistant** and **Oracle Database Configuration Assistant** run after software installation with no user input. Oracle Database Configuration Assistant creates a starter database and adds necessary information to the network configuration. During database creation, Oracle Database Configuration Assistant verifies that the raw devices were created for each tablespace. Raw devices should have been configured in "Setting Up Raw Devices" on page 2-5. If the raw devices are not set up properly, Oracle Database Configuration Assistant fails to create the database.

Custom Installation

Oracle Universal Installer prompts you with a choice to create a database by using one of the following methods:

- Oracle Database Configuration Assistant during installation
- Oracle Database Configuration Assistant or manual methods sometime after installation

This section includes the following topics:

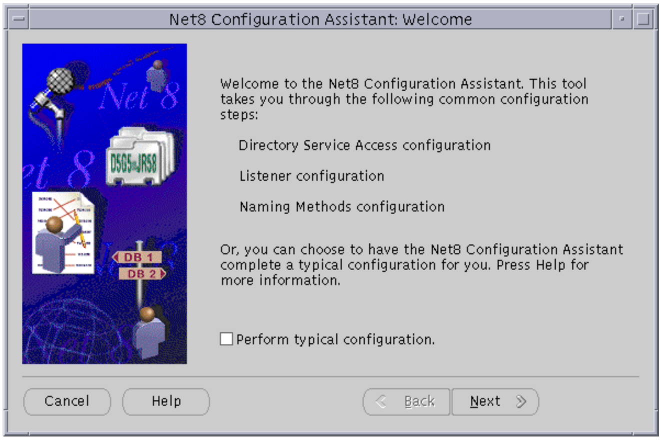
- Net8 Configuration Assistant During Installation
- Oracle Database Configuration Assistant During Installation

> **See Also:** "Creating the Database After Installation" on page 3-23 to create the database after installation

Net8 Configuration Assistant During Installation

Net8 Configuration and Oracle Database Configuration Assistant (depending on your choice) run after software installation

Net8 Configuration Assistant prompts with the following page:



Net8 Configuration Assistant: Welcome

Welcome to the Net8 Configuration Assistant. This tool takes you through the following common configuration steps:

Directory Service Access configuration

Listener configuration

Naming Methods configuration

Or, you can choose to have the Net8 Configuration Assistant complete a typical configuration for you. Press Help for more information.

☐ Perform typical configuration.

Select "Perform typical configuration" to create the network configuration as a Typical installation. Otherwise, follow the prompts to configure directory service access, listener configuration, and naming method configuration.

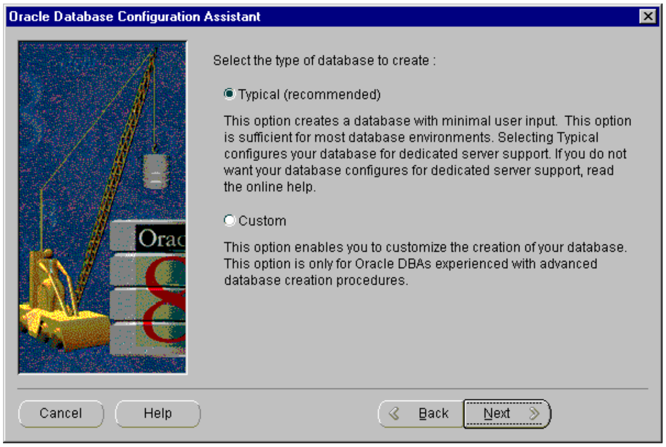
Note: When configuring the listener, Oracle recommends using the TCP/IP protocol on port 1521. TCP/IP is required for Oracle Enterprise Manager.

If you do not use the registered port of 1521, the LOCAL_LISTENER parameter must be set in the `init`*`sid`*`.ora` file and resolved through a naming method. Oracle Database Configuration, which runs after Net8 Configuration Assistant, configures this parameter properly.

For further information about configuring non-default port numbers in the `listener.ora` file, see Chapter 8, "Configuring the Listener," in the *Net8 Administrator's Guide*.

Oracle Database Configuration Assistant During Installation

If you choose to create a database using Oracle Database Configuration Assistant during installation, the following page of the wizard appears:



Oracle Database Configuration Assistant

Select the type of database to create :

● Typical (recommended)

This option creates a database with minimal user input. This option is sufficient for most database environments. Selecting Typical configures your database for dedicated server support. If you do not want your database configures for dedicated server support, read the online help.

○ Custom

This option enables you to customize the creation of your database. This option is only for Oracle DBAs experienced with advanced database creation procedures.

Choose the Typical or Custom database creation type to create a database.

> **See Also:** "Selecting a Database Creation Method" on page 1-10 for a description of the database creation types

Oracle Database Configuration Assistant creates a database and adds necessary information to the network configuration. If directory access was configured with Net8 Configuration Assistant, an entry for the database service is added to the directory. Clients, also configured with directory access, can access the network information for the database service and connect to the database without a `tnsnames.ora` file.

> **See Also:** Chapter 2, "Net8 Concepts," in the *Net8 Administrator's Guide* for further information about LDAP-compliant directory support

During database creation, Oracle Database Configuration Assistant verifies that the raw devices were created for each tablespace. Raw devices should have been configured in "Setting Up Raw Devices" on page 2-5. If the raw devices are not set up properly, Oracle Database Configuration Assistant fails to create the database.

Understanding the Installed Configuration

Together, Net8 Configuration Assistant and Oracle Database Configuration Assistant meet all the requirements for proper database creation and Oracle Enterprise Manager discovery of an Oracle Parallel Server. The following sections describe the configured environment in detail:

- Optimal Flexible Architecture
- Operating-System Specific Configuration
- Initialization Parameter Files (initsid.ora and initdb_name.ora files)
- Listener (listener.ora file)
- Directory Server Access (ldap.ora file)
- Net Service Names (tnsnames.ora file)
- Profile (sqlnet.ora file)

Note: Configuration files are created on all nodes in the cluster.

Optimal Flexible Architecture

Oracle Database Configuration Assistant creates necessary Oracle Parallel Server files, following the **Optimal Flexible Architecture (OFA)** guidelines, whereby database files and administrative files, including initialization parameter files, follow standard naming and placement practices.

See Also: Appendix A to understand the OFA file structure

Operating-System Specific Configuration

The following sections describe operating-system specific configuration issues:

- oratab File on UNIX
- db_name.conf file on UNIX
- Registry Values for Oracle Parallel Server Database on Windows NT
- Service Creation on Windows NT

oratab File on UNIX

An entry for the Oracle Parallel Server database is created in the `oratab` file. Oracle Enterprise Manager uses this file during service discovery to determine if the database is an Oracle Parallel Server and the database name.

Oracle Parallel Server entry has the following syntax:

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

where *db_name* is the database name given to your Oracle Parallel Server database, `$ORACLE_HOME` is the directory path to the database, and `N` indicates the database should not be started at reboot time. A sample entry follows for a database named `op`:

```
OP:/private/system/op:N
```

db_name.conf file on UNIX

A file called `db_name.conf` is created in `$ORACLE_HOME/ops`. Oracle Enterprise Manager uses this file during service discovery to determine which instances run on the nodes.

This file contains the following parameters:

Parameter	Description
`node_list`	Defines the nodes for an Oracle Parallel Server. The node numbers are determined by the Cluster Manager software.
`inst_oracle_sid`	Defines the SID for each instance in the clusters
`lsnr_listener_name`	Defines the listener name for each instance.

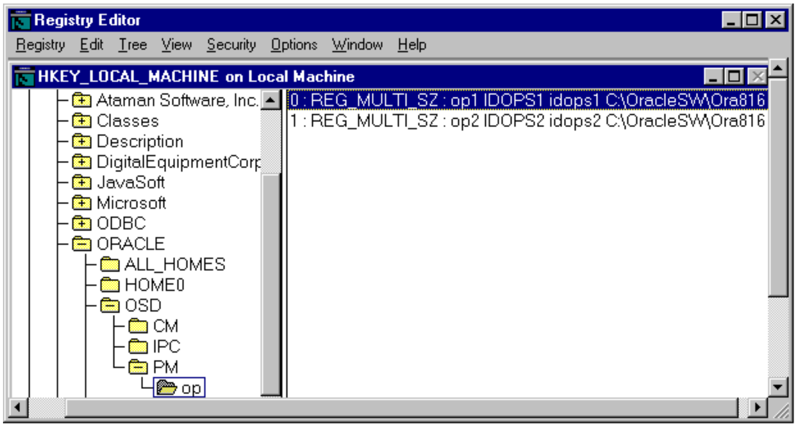
A sample entry follows for a database named `op`:

```
node_list="1-2"
inst_oracle_sid=(op1, op2)
lsnr_listener_name="listener"
```

Registry Values for Oracle Parallel Server Database on Windows NT

On Windows NT, the registry lists all Oracle Parallel Servers that run on a node under the subkey HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\OSD\PM. Under this subkey, each Oracle Parallel Server cluster has its own registry subkey. Oracle Enterprise Manager uses information to discover the Oracle Parallel Server database name, its instances and nodes.

The following Registry Editor graphic shows a subkey for a database named `op`:



```
0 : REG_MULTI_SZ : op1 IDOPS1 idops1 C:\OracleSW\Ora816
1 : REG_MULTI_SZ : op2 IDOPS2 idops2 C:\OracleSW\Ora816
```

op contains entries for two instances named op1 and op2. The registry values names are node numbers for the instances, starting at 0 for the first node. The second node has a sequential value of 1. These registry values are described in the following table:

Value	Value Type	Description
0	REG_MULTI_SZ	Specifies the cluster instance ID data assigned to the op1 on the first node with the following format: *SID COMPUTER_NAME HOST_NAME ORACLE_HOME* `op1 IDOPS1 idops1 c:\OracleSW\Ora816`
1	REG_MULTI_SZ	Specifies the cluster instance ID data assigned to the op2 on the second node with the following format: *SID COMPUTER_NAME HOST_NAME ORACLE_HOME* `op2 IDOPS2 idops2 c:\OracleSW\Ora816`

Service Creation on Windows NT

Each node is configured with an OracleService*sid* service for the instance on that node. It is a Windows NT service that can be controlled from the Control Panel.

This service must be started before the database instance is started.

See Also: "Starting the Database in Parallel Mode" on page 3-41

Initialization Parameter Files (init*sid*.ora and init*db_name*.ora files)

The instance (System Global Area and background processes) for any Oracle database is started using a parameter file. Each node is configured with an instance-specific initialization parameter file, called `init`*`sid`*`.ora`, for the instance and a common initialization parameter file, called `init`*`db_name`*`.ora`, for an Oracle Parallel Server.

Oracle Database Configuration Assistant creates the `init`*`sid`*`.ora` and `init`*`db_name`*`.ora` files. The file names are based on the global database name and SID information entered in the Database Identification page of Oracle Universal Installer.

See Also:

- "Initialization Parameter Files" on page 1-14
- "Understanding the Initialization Parameter Files" on page 4-4 for further information about these files

This section includes the following topics:

- File Location
- Multi-Threaded Server Configuration
- Non-Default Listeners

File Location

For a Typical installation, Oracle Database Configuration Assistant creates the `init`*`sid`*`.ora` and `init`*`db_name`*`.ora` files in `$ORACLE_BASE/admin/`*`db_name`*`/pfile` on UNIX and *`ORACLE_BASE`*`\admin\`*`db_name`*`\pfile` on Windows NT for a starter database.

For a Custom installation, the `init`*`sid`*`.ora` file is created in `$ORACLE_BASE/admin/`*`db_name`*`/pfile` on UNIX and *`ORACLE_BASE`*`\admin\`*`db_name`*`\pfile` on Windows NT for a starter database. The `init`*`db_name`*`.ora` file is also created at this location if the Typical database creation type was chosen. The Custom database creation type enables you to set the `init`*`db_name`*`.ora` file location.

Multi-Threaded Server Configuration

For a Typical installation, the `init`*`db_name`*`.ora` files is configured with **multi-threaded server (MTS)**.

For a Custom installation, MTS is configured if:

- 20 or more users is configured for OLTP database with the Typical database creation type
- Shared Server Mode is selected for a Custom database creation type

MTS is configured in the following manner:

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

This configuration enables **connection load balancing**, whereby the number of active connections is balanced among the various instances and MTS **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. `listeners_`*`db_name`* is resolved to listener addresses through a `tnsnames.ora` file.

See Also:

- "Net Service Names (tnsnames.ora file)" on page 3-16
- *Net8 Administrator's Guide* for further information about the MTS_DISPATCHERS parameter.

Non-Default Listeners

If you configured a listener that does not use the default listening address—TCP/IP, port 1521—the LOCAL_LISTENER parameter in the `init`*`sid`*`.ora` file is automatically configured as follows:

```
local_listener=listener_sid
```

where `listener_`*`sid`* is resolved to a listener address through either a `tnsnames.ora` file on the machine.

Multiple Listeners

If Oracle Database Configuration Assistant detects more than one listener on each of the nodes, it displays a list of the listeners. You can select one the listeners displayed. If you select a non-default listener, the LOCAL_LISTENER parameter is set in the `init`*`sid`*`.ora` file, as described in "Non-Default Listeners".

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 Typical install, Net8 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 Custom installation, you are prompted to create at least one listener with Net8 Configuration Assistant. The listener is configured to listen on one protocol address you specify, as well as an address for **external procedures**.

Both installation modes configure service information about the Oracle Parallel Server database and external procedures. An Oracle release 8.1 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 management tools require service configuration in the `listener.ora` file to discover the database. The database service information includes the **global database name** of the database and Oracle System Identifier (SID) information of the instance.

After listener creation, the listener is started by Net8 Configuration Assistant.

A sample `listener.ora` file with an entry for an instance named `op1` follows:

```
listener=
  (description=
    (address=(protocol=ipc)(key=extproc)))
    (address=(protocol=tcp)(host=op1-sun1)(port=1521)))
sid_list_listener=
  (sid_list=
    (sid_desc=
      (sid_name=plsextproc)
      (oracle_home=/orahome81)
      (program=extproc)
     (sid_desc=
       (oracle_home=/orahome81)
       (sid_name=op1)))
```

Notice that the second SID_DESC entry for the instance does not use have GLOBAL_DBNAME parameter entry. 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/8.1.6)))
```

In Oracle Parallel Server environment, the GLOBAL_DBNAME parameter disables **connect-time failover** or **Transparent Application Failover (TAF)**, Oracle Corporation strongly recommends that you not add this parameter to the `listener.ora` file.

> **See Also:** *Net8 Administrator's Guide* for further information about the listener and the `listener.ora` file

Directory Server Access (ldap.ora file)

If you chose to configure access to a LDAP-compliant directory server with Net8 Configuration Assistant during a Custom installation, a `ldap.ora` file is created. This file contains the following types of information:

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

> **See Also:** *Net8 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 a centralized directory (if configured during a Custom 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 a Oracle release 8.1 database or SID for an Oracle release 8.0 or version 7 database

Net service names are created for the following connections:

<table>
<tr><th>Net Service Name Type</th><th>Description</th></tr>
<tr><td>Database connections</td><td>
<p>Oracle Enterprise Manager searches for a net service name entry for the database. This entry enables Oracle Enterprise Manager to discover an Oracle Parallel Server database and to determine which instances to use for a connection.</p>
<p>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.</p>
<p>In the following example, <code>op.us.acme.com</code> is used by Oracle Enterprise Manager to connect to the target database, <code>op.us.acme.com</code>.</p>
<pre>
op.us.acme.com=
 (description=
  (load_balance=on)
   (address=(protocol=tcp)(host=op1-sun)(port=1521)
   (address=(protocol=tcp)(host=op2-sun)(port=1521)
  (connect_data=
     (service_name=op.us.acme.com)))
</pre>
<p><b>Note:</b> FAILOVER=ON in set by default for a list of addresses. Thus, it does need to be explicitly specified.</p>
</td></tr>
<tr><td>Instance connections</td><td>
<p>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.</p>
<p>In the following example, <code>op1.us.acme.com</code>, is used by Oracle Enterprise Manager to connect to an instance named <code>op1</code> on <code>op1-server</code>:</p>
<pre>
op1.us.acme.com=
 (description=
  (address=(protocol=tcp)(host=op1-server)(port=1521))
  (connect_data=
    (service_name=op.us.acme.com)
    (instance_name=op1)))
</pre>
</td></tr>
</table>

<table>
<tr><th>Net Service Name Type</th><th>Description</th></tr>
<tr><td>Startup instance connections for Windows NT</td><td>

If Oracle Parallel Server is installed on Windows NT, additional net service name entries are created with connections to the SID rather than the database service name. These net service names are identified uniquely by `*sid*_startup` for each instance. These entries enable Oracle Enterprise Manager to start database instances remotely on Windows NT.

On Windows NT, a remote connection is performed from one of the nodes to the other nodes. This remote connection cannot accept the SERVICE_NAME and INSTANCE_NAME parameters if the database is down. Therefore, a remote connection must specify the SID to start in dedicated server mode.

In the following example, `op1_startup.us.acme.com` is used by Oracle Enterprise Manager to start the `op1` instance.

```
op1_startup.us.acme.com=
 (description=
  (address=(protocol=tcp)(host=op1-pc)(port= 1521))
  (connect_data=
    (sid=op1)
    (server=dedicated))
```

When Oracle Enterprise Manager performs starts an instance on a UNIX node, it passes a command to a process on the node to perform a local connection. Therefore, these entries are not created.

</td></tr>
<tr><td>Remote listeners</td><td>

As discussed in "Multi-Threaded Server Configuration" on page 3-13, the MTS_DISPATCHERS parameter is set in the `init*sid*.ora` file with the LISTENER attribute:

```
mts_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 for the first three nodes in the cluster through the `tnsnames.ora` file.

In the following example, `listeners_op.us.acme.com` is resolved to list of listeners available in the cluster:

```
listeners_op.us.acme.com=
   (address=(protocol=tcp)(host=op1-sun)(port= 1521))
   (address=(protocol=tcp)(host=op2-sun)(port=1521))
```

The instance uses this list to determine the names of the remote listeners to register its information.

</td></tr>
</table>

Net Service Name Type	Description

Non-default listeners	As discussed in "Non-Default Listeners" on page 3-13 and "Multiple Listeners" on page 3-13, the LOCAL_LISTENER parameter is set in the `init`*`sid`*`.ora` file if a non-default listener is configured.
	`local_listener=listener_`*`sid`*
	`listener_`*`sid`* is resolved to a listener address.
	In the sample below, `listener_op1.us.acme.com` is resolved to the non-default listener address:
	`listener_op1.us.acme.com=`
	`   (address=(protocol=tcp)(host=op1-sun)(port= 1421))`

External procedures	An entry for connections to **external procedures**. This enables an Oracle8*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)))`

A sample tnsnames.ora file created during a Typical installation follows:

```
op.us.acme.com=
 (description=
  (load_balance=on)
  (failover=on)
  (address_list=
   (address=(protocol=tcp)(host=op1-sun)(port=1521))
   (address=(protocol=tcp)(host=op2-sun)(port=1521)))
  (connect_data=
     (service_name=op.us.acme.com)))

op1.us.acme.com=
 (description=
  (address=(protocol=tcp)(host=op1-sun)(port=1521))
  (connect_data=
    (service_name=op.us.acme.com)
    (instance_name=op1)))

op2.us.acme.com=
 (description=
  (address=(protocol=tcp)(host=op2-sun)(port=1521))
  (connect_data=
    (service_name=op.us.acme.com)
    (instance_name=op2)))

listeners_op.us.acme.com=
   (address=(protocol=tcp)(host=op1-sun)(port= 1521))
   (address=(protocol=tcp)(host=op2-sun)(port=1521))

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

See Also: *Net8 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, op in the connect string CONNECT scott/tiger@op gets looked up as op.us.acme.com.

- Naming methods the server can use to resolve a name to a **connect descriptor**

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

A sample sqlnet.ora file created during a Typical installation follows:

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

See Also: *Net8 Administrator's Guide* for further information about the sqlnet.ora file

Migrating or Upgrading to Release 8.1

If Oracle Universal Installer detects an earlier version of an Oracle database on your hard drive, you are prompted to migrate or upgrade the database to release 8.1.

Important: Do not click the "Migrate an Existing Database" check box, as Oracle Data Migration Assistant does not support Oracle Parallel Server. Instead, use the Migration utility to migrate after installation, as described in *Oracle8i Migration* guide.

Note: Migrating or upgrading on Windows NT involves disabling the OraclePGMSService and starting and shutting down the Operating System Dependent layer at a specific point during migration. These instructions are not covered in the *Oracle8i Migration* guide. See *Oracle8i Administrator's Guide for Windows NT* for complete migration and upgrade steps.

Coexistence

As long as your Oracle database software version numbers are greater than 8.1, they can co-exist on the same cluster. This also 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 Oracle Parallel Server database are not compatible on the same cluster.

Rolling Upgrades

Rolling upgrades—where different databases, or different instances of the same database in Oracle Parallel Server, are upgraded to a new version of Oracle one at a time—are not supported.

Multiple Oracle Homes

Like an Oracle8*i* Enterprise Edition database, Oracle Parallel Server on UNIX supports multiple Oracle homes, a feature that enables you to install one or more releases on the same machine (in multiple Oracle home directories). Windows NT also supports multiple Oracle homes, but all the Oracle homes must contain the same release of Oracle8*i* Enterprise Edition and Oracle Parallel Server.

Both UNIX and Windows NT require that each node have only one copy of the vendor-supplied Operating System Dependent layer active.

Creating the Database After Installation

If you choose not to create a database during installation, you can create a database using Oracle Database Configuration Assistant or manuals methods. Specific topics discussed are:

- Use Oracle Database Configuration Assistant in Stand-Alone Mode
- Use Manual Methods

Use Oracle Database Configuration Assistant in Stand-Alone Mode

This section describes using Oracle Database Configuration Assistant in stand-alone mode:

- Prerequisite Requirement
- Create the Database

Prerequisite Requirement

In order to create a database with Oracle Database Configuration Assistant, the raw devices for the tablespaces must be created. During database creation, Oracle Database Configuration Assistant verifies that the raw devices were created for each tablespace. If the raw devices are not set up properly, Oracle Database Configuration Assistant fails to create the database.

See Also: "Setting Up Raw Devices" on page 2-5

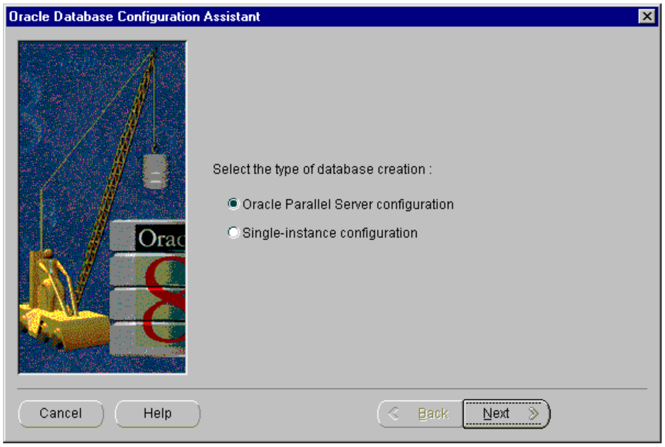
Create the Database

To create a database with Oracle Database Configuration Assistant:

1. Start Oracle Database Configuration Assistant on one of the nodes:

   - On Windows NT, choose Start > Programs > Oracle for Windows NT - [*HOME_NAME*] > Oracle Database Configuration Assistant.
   - On UNIX, run `dbassist` from `$ORACLE_HOME/bin`.

   The welcome page appears:



Oracle Database Configuration Assistant

Select the type of database creation :

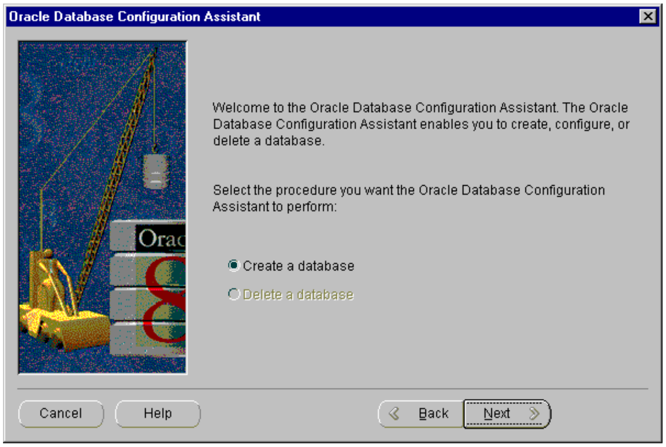
● Oracle Parallel Server configuration

○ Single-instance configuration

> **Note:** If this page does not display, it means Oracle Database Configuration Assistant was unable to:
>
> - Detect the Lock Manager software or the lists of nodes in the cluster on UNIX operating systems
> - Load the Cluster Manager software on Windows NT
>
> See your Operating System Dependent vendor documentation to resolve the problem, then start Oracle Database Configuration Assistant again.

2. Select Oracle Parallel Server Configuration, then click Next.

The following page appears:



Oracle Database Configuration Assistant

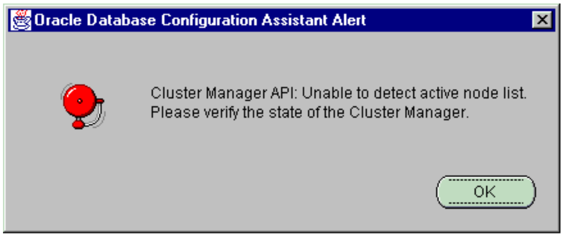
Welcome to the Oracle Database Configuration Assistant. The Oracle Database Configuration Assistant enables you to create, configure, or delete a database.

Select the procedure you want the Oracle Database Configuration Assistant to perform:

● Create a database

○ Delete a database

The following error message may appear if there is a problem with the clusterware:



Oracle Database Configuration Assistant Alert

Cluster Manager API: Unable to detect active node list. Please verify the state of the Cluster Manager.

On UNIX, this message indicates that Oracle Database Configuration Assistant is unable to detect the lists of nodes in the cluster. See your Operating System Dependent vendor documentation for further information.

On Windows NT, this messages indicates that Cluster Manager (CM) is installed but one of the following problems were detected:

– There is an error calling the API

– The local node is not recognized

– You do not have administrative privilege to any of the nodes

To resolve this error message:

a. Check whether CM was properly installed and configured. See your vendor's Operating System Dependent layer documentation for further information.

b. Check if you have administrative privileges on nodes by entering:

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

where *host_name* is the host name defined in the DefinedNodes registry value for CM.

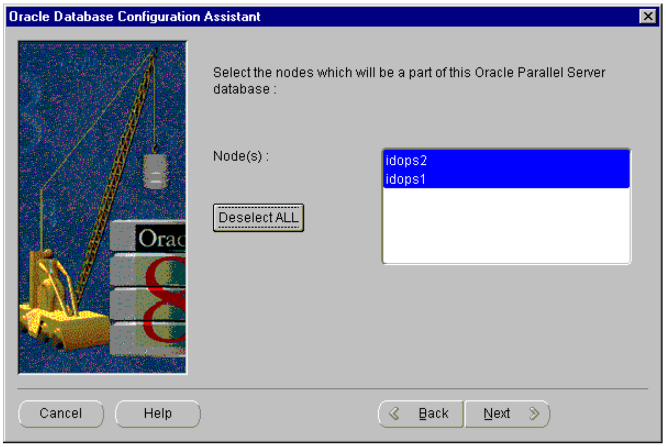
A successful connection results in the following message:

```
The command completed successfully.
```

Oracle recommends using the same user name and password on each node in a cluster or using a domain user name. If you use a domain user name, log on under a domain with username and password which has administrative privileges on each node.

3. Click Create a database, then click Next.

The following page appears:



Oracle Database Configuration Assistant

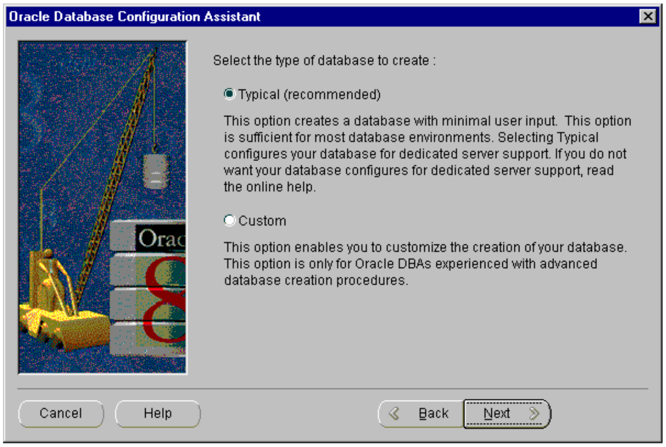
Select the nodes which will be a part of this Oracle Parallel Server database :

Node(s) :

idops2
idops1

4. Select the nodes where you want the database created.

The following page appears:



Oracle Database Configuration Assistant

Select the type of database to create :

● Typical (recommended)

This option creates a database with minimal user input. This option is sufficient for most database environments. Selecting Typical configures your database for dedicated server support. If you do not want your database configures for dedicated server support, read the online help.

○ Custom

This option enables you to customize the creation of your database. This option is only for Oracle DBAs experienced with advanced database creation procedures.

5. Choose the Typical or Custom database creation type to create a database.

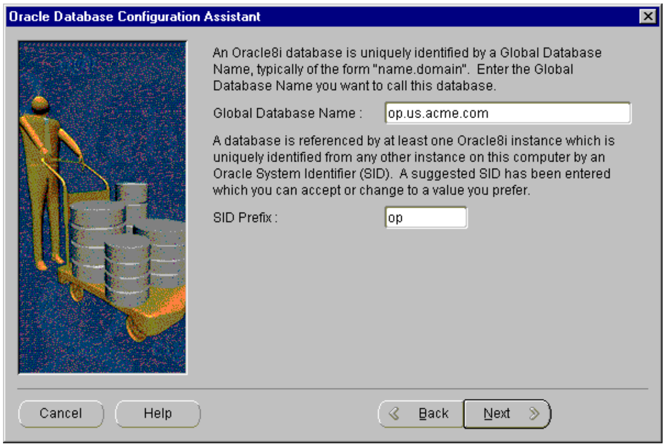
   See Also: "Selecting a Database Creation Method" on page 1-10 for a description of these database creation types

6. Respond to instructions in each Oracle Database Configuration Assistant page, then click Next when you are ready to continue to the next page.

   When you get to the page that requests the **global database name** and SID Prefix:

   - Enter an appropriate **global database name**, a name comprised of the database name and database domain, such as `op.us.acme.com`.
   - Accept or change the prefix for the **Oracle System Identifier (SID)** in the SID Prefix field.

The prefix entered here is appended with each node's thread ID to create the SID for instances. For example, if `op` is entered, the first instance in the cluster is given a SID of `op1`, and the second instance is given a SID of `op2`.



Oracle Database Configuration Assistant

An Oracle8i database is uniquely identified by a Global Database Name, typically of the form "name.domain". Enter the Global Database Name you want to call this database.

Global Database Name : op.us.acme.com

A database is referenced by at least one Oracle8i instance which is uniquely identified from any other instance on this computer by an Oracle System Identifier (SID). A suggested SID has been entered which you can accept or change to a value you prefer.

SID Prefix : op

7. When you get to the last screen, click Finish to start the creation of the Oracle Parallel Server database.

In addition to database creation, Oracle Database Configuration Assistant adds necessary information to the network configuration. If directory access was configured with Net8 Configuration Assistant, an entry for the database service is added to the directory. Clients, also configured with directory access, can access the network information for the database service and connect to the database without a `tnsnames.ora` file.

See Also: "Understanding the Installed Configuration" on page 3-8 to understand how the configuration files are configured

Use Manual Methods

You might choose to create your database manually if you already have existing scripts, or have different requirements than can be met by using Oracle Database Configuration Assistant.

To create a new database and make it available for system use, perform the following tasks:

Task 1: Specify database and instance settings.

Task 2: Back up any existing databases.

Task 3: Set operating-system specific configuration.

Task 4: Set ORACLE_SID for each node.

Task 5: Create initialization parameter files.

Task 6: Create the password files.

Task 7: Prepare a CREATE DATABASE OPS script.

Task 8: Create the database.

Task 9: Back up the database.

Task 10: Configure Net8 on nodes.

In order to understand the database creation process, you may want to review all the steps before performing the tasks.

Task 1: Specify database and instance settings.

In Oracle Parallel Server, each node has its own instance. The instances together form an Oracle Parallel Server database. Being aware of database- and instance-level information enables you to more easily complete Task 2 through Task 12.

To determine database and instance-level information:

1. Determine the following settings for your database:

Component	Description
Database Name	The name of your database
Database Domain	The domain name
Global Database Name	A name comprised of database name and database domain.
SID Prefix	A prefix for the **Oracle System Identifier (SID)**. The prefix is appended to the **thread ID** for the node to create the SID for node's instance.

2. Create a table that lists the settings. For example, the following table shows the settings for a database named `op`:

Database Name	Database Domain	Global Database Name	SID Prefix
`op`	`us.acme.com`	`op.us.acme.com`	`op`

3. Determine the following settings for each node:

Component	Description
Node name	The node name defined by the Cluster Manager software Use the command `lsnodes -l -n` to obtain the computer's node name. `lsnodes` is located in the `$ORACLE_HOME/bin` directory on UNIS and `ORACLE_HOME\bin` directory on Windows NT.
Node number	The number associated with the node. Use the command `lsnodes -l -n` to obtain the node number of the node. On Windows NT, the first nodes always has a node number of 0. Other nodes should have sequential values, such as 1 for the second node, and 2 for the third node, and so on.

Component	Description
Host name	The host name of the computer. The host name may be the same names as the node name. On UNIX and Windows NT, use the command `hostname` to obtain the host name for the computer.
Oracle Home	The directory path location where Oracle Universal Installer installed the software, including binaries
Oracle Base	The directory path location set up by Oracle Universal Installer. The Oracle Base contains the Oracle Home. It can also be used to store administrative files, including initialization parameter files and database trace files. Oracle Corporation recommends using the OFA directory structure used by the Oracle Database Configuration Assistant, which places administrative files under a directory called `$ORACLE_BASE\admin` on UNIX and *`ORACLE_BASE/`*`admin` on Windows NT. **See Also:** "Understanding the Oracle Parallel Server Directory Structure" on page A-2
Thread ID	Each node requires a unique **thread ID**. The thread ID is appended to the SID prefix to create the SID for the instance on the node.

4. Create a table that lists the settings. For example, the following table shows the settings for a nodes named `idops1` and `idops2`:

Node Name	Node Number	Host Name	Oracle Base	Oracle Home	Thread ID	SID
idops1	0	idops1	C:\OracleSW	C:\OracleSW\Ora816	1	op1
idops2	1	idops2	C:\OracleSW	C:\OracleSW\Ora816	2	op2

Task 2: Back up any 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, and control files.

See Also: *Oracle8i Backup and Recovery Guide*

Task 3: Set operating-system specific configuration.

The following sections describe the operating-system specific configuration:

- oratab File on UNIX
- db_name.conf File on UNIX
- Registry Values for Oracle Parallel Server Database on Windows NT

oratab File on UNIX If you plan to use Oracle Enterprise Manager, manually create an entry in `oratab` file on each node. This file identifies the Oracle Parallel Server database. Oracle Enterprise Manager uses this file during service discovery to determine if the database is an Oracle Parallel Server and its database name.

The `oratab` file is stored in `/etc/oratab` or `/var/opt/oracle/oratab`, depending on your operating system.

The syntax for this entry is:

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

db_name is the database name given to your Oracle Parallel Server database, `$ORACLE_HOME` is the directory path to the database, and `N` indicates the database should not be started at reboot time. Use the database name and Oracle home you specified in Task 1: Specify database and instance settings.

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

```
OP:/private/system/op:N
```

***db_name*.conf File on UNIX** If you plan to use Oracle Enterprise Manager, manually create the `db_name.conf` file on each node. Ensure that it contains the `node_list`, `inst_oracle_sid`, and `lsnr_listener_name` parameters. Oracle Enterprise Manager searches for these entries to discover what instance and listener run on each node. The `db_name.conf` file must be stored in `$ORACLE_HOME/ops`. If the `/ops` directory does not exist, create it.

Sample contents follow for a two-node cluster:

```
node_list="0,1"
inst_oracle_sid=(op1, op2)
lsnr_listener_name="listener"
```

Any value can be used for the `lsnr_listener_name` parameter. Use the node number for the `node_list` parameter and the SID value for the `inst_oracle_sid` parameter you specified in Task 1: Specify database and instance settings.

> **See Also:** "Parameter Descriptions" on page B-4 for a description of these parameters and their use

Registry Values for Oracle Parallel Server Database on Windows NT If you plan to use Oracle Enterprise Manager, create a \PM*db_nam*e key in HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\OSD for the Oracle Parallel Server database. Oracle Enterprise Manager uses this key to discover the Oracle Parallel Server database name, its instances and nodes.

> **See Also:** *Oracle8i Administrator's Guide for Windows NT* for further information about creating registry values

The PM and *db_name* keys should have a class type of REG_SZ.

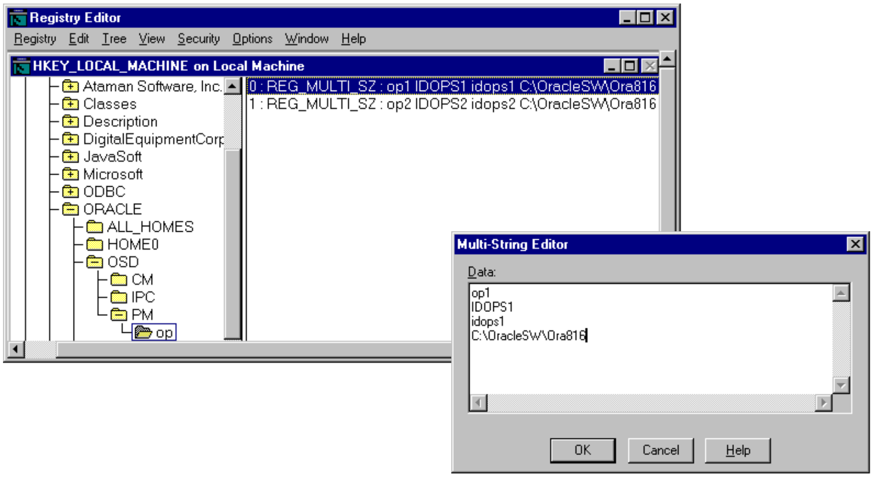
The *db_name* key must contain registry values for each instance in the cluster. The registry values names are node numbers for the instances, starting at 0 for the first node, regardless of the node's thread ID. Other nodes should have sequential values, such as 1 for the second node, and 2 for the third node, and so on.

When creating the registry values, the data type must be REG_MULTI_SZ. In the Multi-String Editor dialog box, enter the following on separate lines:

- SID of the node
- Computer name of the machine
- Host name of the machine
- Oracle home directory

Use the information you entered in Task 1: Specify database and instance settings.

The following Registry Editor graphic shows a subkey of op with values entered for instance op1:



0 : REG_MULTI_SZ : op1 IDOPS1 idops1 C:\OracleSW\Ora816
1 : REG_MULTI_SZ : op2 IDOPS2 idops2 C:\OracleSW\Ora816

```
op1
IDOPS1
idops1
C:\OracleSW\Ora816
```

op contains two registry values for two instances, as described in the following table:

Value	Value Type	Description
0	REG_MULTI_SZ	Specifies the cluster instance ID data assigned to the op1 on the first node with the following format: *SID COMPUTER_NAME HOST_NAME ORACLE_HOME* `op1 IDOPS1 idops1 c:\OracleSW\Ora816`
1	REG_MULTI_SZ	Specifies the cluster instance ID data assigned to the op2 on the second node with the following format: *SID COMPUTER_NAME HOST_NAME ORACLE_HOME* `op2 IDOPS2 idops2 c:\OracleSW\Ora816`

Task 4: Set ORACLE_SID for each node.

The SID must be defined for each node in the cluster. For simplicity, Oracle recommends SIDs that consist of the database name as the common base and the thread ID of the node you specified in Task 1: Specify database and instance settings. For example, if `op` is the database name, the first instance in the cluster is given a SID of `op1`, and the second instance is given a SID of `op2`. SID specification is operating-system specific:

- UNIX
- Windows NT

UNIX On UNIX, the ORACLE_SID environment variable must be set.

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

Windows NT On Windows NT, the ORACLE_SID registry value must be set in HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*ID*.

> **See Also:** *Oracle8i Administrator's Guide for Windows NT* for further information about this registry value

Once the SIDs are created, 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.

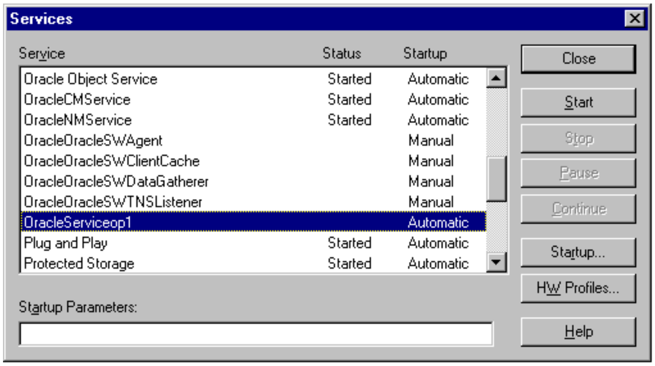
   ```
   C:\ORACLE_HOME\bin> crtsrv sid
   ```

   For example, to create a service for a SID of `op1`, OracleServiceop1, enter the following:

   ```
   C:\ORACLE_HOME\bin> crtsrv op1
   ```

2. Verify OracleService*sid* exists by choosing the Services icon from the Control Panel.

The Services window appears:



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

Startup Parameters:

Task 5: Create initialization parameter files.

The instance (System Global Area and background processes) for any Oracle database is started using a parameter file. For each node, Oracle recommends creating an instance-specific initialization parameter file, named `init`*`sid`*`.ora`, for the instance parameters and a common initialization parameter file, named `init`*`db_name`*`.ora`, for the Oracle Parallel Server parameters.

To create a parameter file for the database you are about to make, make a copy of the initialization parameter files located in `$ORACLE_HOME/opsm/admin` on UNIX or *`ORACLE_HOME`*`\opsm\admin` on Windows NT. Give these files new file names. You can then edit and customize these new files for the new database.

To create a new database, inspect and edit the following parameters of the new `init`*`sid`*`.ora` parameter file:

- IFILE
- INSTANCE_NAME
- INSTANCE_NUMBER
- ROLLBACK_SEGMENTS
- THREAD
- REMOTE_LOGIN_PASSWORDFILE

Inspect and edit the following parameters of the new `init`*`db_name`*`.ora` parameter file:

- BACKGROUND_DUMP_DEST
- CONTROL_FILES
- DB_DOMAIN
- DB_NAME
- MTS_DISPATCHERS
- SERVICE_NAMES
- USER_DUMP_DEST

> **See Also:** "Understanding the Initialization Parameter Files" on page 4-4 for further information about initialization parameter files and parameters to use

Task 6: Create the password files.

Use the Password Utility ORAPWD to create password files. ORAPWD is automatically installed with the Oracle8*i* Utilities. Password files are located in the `$ORACLE_HOME/dbs` on UNIX and *`ORACLE_HOME`*`\database` directory on Windows NT. They are named `orapw`*`sid`* on UNIX and `pwd`*`sid`*`.ora` on Windows NT, where *sid* identifies the database name you specified in Task 1: Specify database and instance settings.

To create a password file on each of the nodes:

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/orapwsid password=password
     ```

   - On Windows NT, 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 INTERNAL and SYS accounts.

2. Ensure instance initialization file parameter REMOTE_LOGIN_PASSWORDFILE is set EXCLUSIVE.

Task 7: Prepare a CREATE DATABASE OPS script.

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

1. Set PFILE so it points to the location of the `init`*`db_name`*`.ora` file.
2. Modify `oracle` in the `CONNECT INTERNAL\oracle` line to reflect the password you created in Task 6: Create the password files.
3. Modify the location of the data dictionary scripts, `$ORACLE_HOME/rdbms/admin` on UNIX and *`ORACLE_HOME`*`\rdms/admin` on Windows NT, to reflect the Oracle home you specified in Task 1: Specify database and instance settings.
4. Modify the log file and data file names with the file names or symbolic link names you created in "Setting Up Raw Devices" on page 2-5.
5. Modify the log file and data file sizes for the Oracle Parallel Server.
6. Create enough private (acquired explicitly by an instance when an instance opens a database) rollback segments (two per node is required) for the number of concurrent users per transaction. With the exception of the SYSTEM rollback segment, public rollback segments cannot be shared among nodes.

Task 8: Create the database.

To create the new database, run the SQL CREATE DATABASE OPS script:

```
SQL> @path/ops.sql;
```

When you execute this script, Oracle performs the following operations:

- Creates the data files for the database
- Creates the control files for the database
- Creates the redo log files for the database
- Creates the data dictionary
- Creates the SYSTEM tablespace and the SYSTEM rollback segment
- Creates the users SYS and SYSTEM
- Mounts and opens the database for use

Task 9: Back up the database.

You should make a full backup of the database to ensure that you have a complete set of files from which to recover if a media failure occurs.

See Also: *Oracle8i Backup and Recovery Guide*

Task 10: Configure Net8 on nodes.

The `listener.ora` file, `sqlnet.ora` file and `tnsnames.ora` files must be configured properly, as described in the following table:

Configuration File	Description	Configuration Requirements
`listener.ora`	Includes addresses of all network listeners on a server, the SIDs 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 node in the SID_LIST_*listener_name* section. **See Also:** ■ "Listener (listener.ora file)" on page 3-14 for a sample configuration file ■ Chapter 7, "Configuring the Listener," in the *Net8 Administrator's Guide* for configuration procedures
`tnsnames.ora`	Includes a list of network descriptions of service names, called net service names	**See Also:** ■ "Net Service Names (tnsnames.ora file)" on page 3-16 for net service name requirements ■ Chapter 6, "Configuring Naming Methods," in the *Net8 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 3-21 for a sample configuration

See Also: *Net8 Administrator's Guide* to create these files

Starting the Database in Parallel Mode

To start the Oracle Parallel Server database in parallel mode:

> **Note:** Prior to performing the following tasks, ensure your vendor-supplied CM component is configured and started on each node. See your Operating System Dependent vendor documentation for further information.

1. On Windows NT only, start OracleService*sid* instance on *each* node.

   - From the MS-DOS command line, enter:

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

   - From the Control Panel's Services window, select OracleService*sid*, then click Start.

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

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

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

   LSNRCTL displays a status message indicating that the listener has started successfully. Check that all expected services for that listener are listed in the services summary in the status message.

3. On one of the nodes, start the database:

   ```
   SQL> CONNECT internal/password
   SQL> STARTUP;
   ```

   The first instance to start up in shared mode determines the values of the **global cache** parameters for the other instances. The control file records the values of the GC_* parameters when the first instance starts up.

   When another instance attempts to start up in shared mode, the Oracle Parallel Server compares the values of the global constant parameters in its parameter file with those already in use and issues a message if any values are incompatible. The instance cannot mount the database unless it has the correct values for its global constant parameters.

4. On the remaining nodes, start the database:

```
SQL> CONNECT internal/password
SQL> STARTUP;
```

Verifying Instances Are Running

To verify instances are running:

1. On any node, enter the following:

```
SQL> CONNECT internal/password
SQL> SELECT * from v$active_instances;
```

Output similar to the following is returned.

```
INST_NUMBER INST_NAME
-----------  -----------------
          1 op1-sun:op1
          2 op2-sun:op2
          3 op3-sun:op3
```

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

2. On all the nodes, connect as SCOTT/TIGER, and view the EMP table.

```
SQL> CONNECT scott/tiger
SQL> SELECT * from emp;
```

The employee table displays:

```
    EMPNO ENAME      JOB             MGR HIREDATE        SAL      COMM    DEPTNO
--------- ---------- --------- --------- --------- --------- --------- ---------
     7369 SMITH      CLERK          7902 17-DEC-80       800                  20
     7499 ALLEN      SALESMAN       7698 20-FEB-81      1600       300        30
     7521 WARD       SALESMAN       7698 22-FEB-81      1250       500        30
     7566 JONES      MANAGER        7839 02-APR-81      2975                  20
     7654 MARTIN     SALESMAN       7698 28-SEP-81      1250      1400        30
     7698 BLAKE      MANAGER        7839 01-MAY-81      2850                  30
     7782 CLARK      MANAGER        7839 09-JUN-81      2450                  10
     7788 SCOTT      ANALYST        7566 19-APR-87      3000                  20
     7839 KING       PRESIDENT           17-NOV-81      5000                  10
```

```
 7844 TURNER     SALESMAN       7698 08-SEP-81      1500         0        30
 7876 ADAMS      CLERK          7788 23-MAY-87      1100                  20
 7900 JAMES      CLERK          7698 03-DEC-81       950                  30
 7902 FORD       ANALYST        7566 03-DEC-81      3000                  20
 7934 MILLER     CLERK          7782 23-JAN-82      1300                  10
14 rows selected.
```

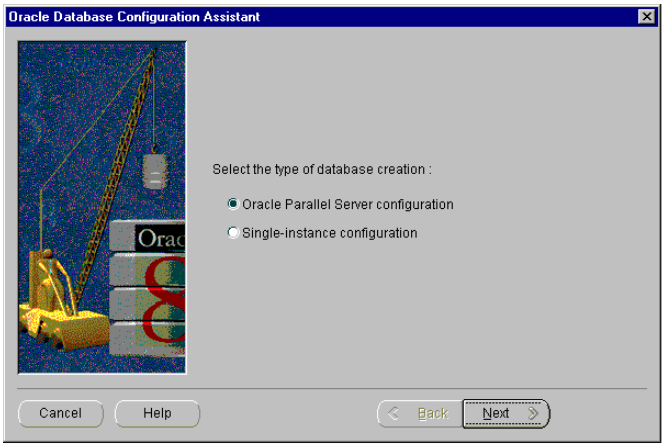
Deleting the Database

The Delete a Database option of Oracle Database Configuration Assistant lets you quickly and easily delete a database. Deleting a database removes its initialization parameter files, instances, OFA structure, and operating system configuration. Data files are not removed from the raw partitions.

To delete a database with Oracle Database Configuration Assistant.

1. Start Oracle Database Configuration Assistant on one of the nodes:
   - On Windows NT, choose Start > Programs > Oracle for Windows NT - [*HOME_NAME*] > Oracle Database Configuration Assistant.
   - On UNIX, run `dbassist` from `$ORACLE_HOME/bin`.

The welcome page appears:



Select the type of database creation :

● Oracle Parallel Server configuration

○ Single-instance configuration

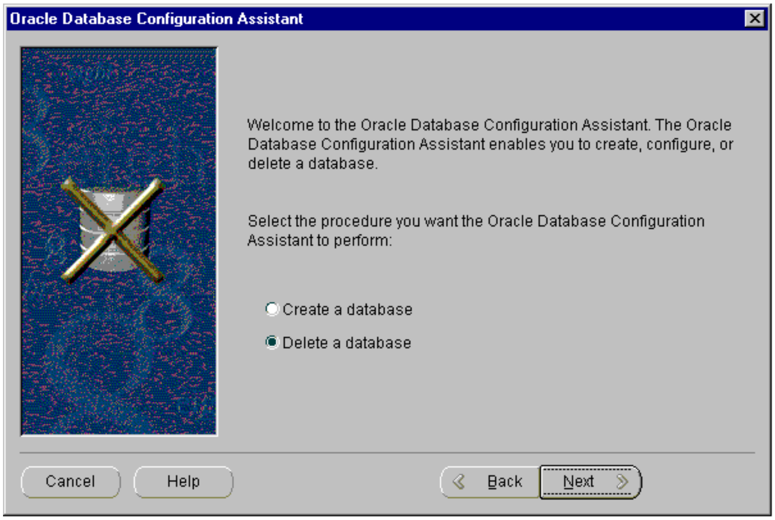
Note: If this page does not display, it means Oracle Database Configuration Assistant was unable to:

- Detect the Lock Manager software or the lists of nodes in the cluster on UNIX operating systems
- Load the Cluster Manager software on Windows NT

See your Operating System Dependent vendor documentation to resolve the problem, then start Oracle Database Configuration Assistant again.

2. Select Oracle Parallel Server Configuration, then click Next.

The following page appears:



Welcome to the Oracle Database Configuration Assistant. The Oracle Database Configuration Assistant enables you to create, configure, or delete a database.

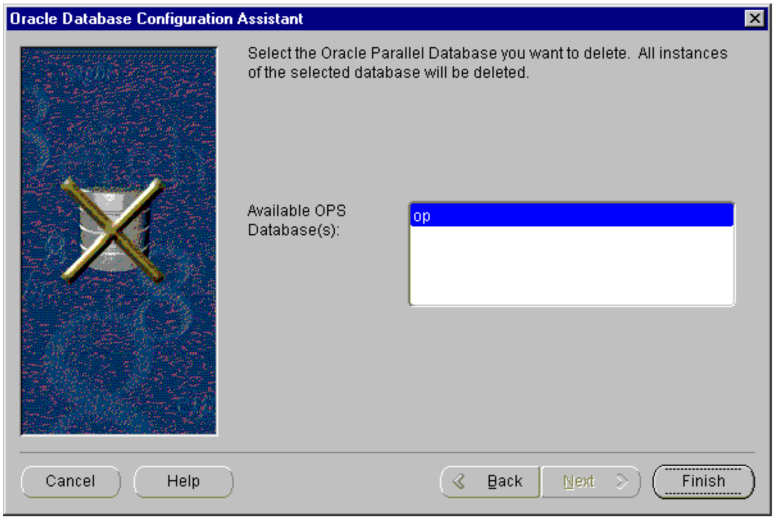
Select the procedure you want the Oracle Database Configuration Assistant to perform:

○ Create a database

● Delete a database

3. Click Delete a database, then click Next.

The following page appears. It displays the Oracle Parallel Server databases detected.



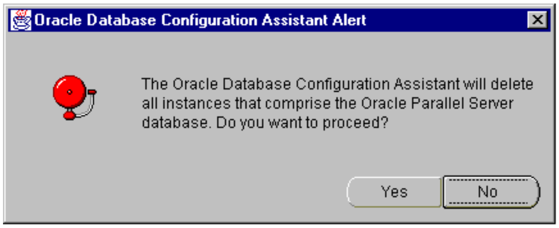
Oracle Database Configuration Assistant

Select the Oracle Parallel Database you want to delete. All instances of the selected database will be deleted.

Available OPS Database(s): op

4. Select the database you want to delete, then click Next. All associated instances for this database are also deleted.

   The following alert dialog box appears:



Oracle Database Configuration Assistant Alert

The Oracle Database Configuration Assistant will delete all instances that comprise the Oracle Parallel Server database. Do you want to proceed?

5. Click Yes to continue with deletion of the database and its associated files, services, and environment settings.

4

Additional Configuration Issues

This chapter describe additional configuration issues not covered by the database creation process.

Specific topics discussed are:

- Configuring Clients for Oracle Parallel Server
- Understanding the Initialization Parameter Files
- Configuring Recovery Manager for Backup and Recovery
- Making a Consistent Backup

Configuring Clients for Oracle Parallel Server

The client should be configured with a net service name for the database. This entry should have an address list of all the listeners in the cluster. Additionally, the connect-time failover and client load balancing options should be set.

Connect-time failover instructs the client to failover to the next listener in the address list if the first one fails. **Client load balancing** instructs the client to randomly select a listener address. This randomization serves to distribute the load so as not to overburden a single listener. Together, these options instruct the client to choose an address randomly. If the chosen address fails, the connection request is failed over to the next address. This way, if an instance should go down, the client can still connect by way of another instance.

Implementation

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:

```
op.us.acme.com=
 (description=
  (load_balance=on)
  (failover=on)
  (address=(protocol=tcp)(host=idops1)(port=1521))
  (address=(protocol=tcp)(host=idops2)(port=1521))
  (connect_data=
     (service_name=op.us.acme.com)))
```

See Also:

- Chapter 6, "Configuring Naming Methods," in the *Net8 Administrator's Guide* to configure a connect descriptor
- Chapter 8, "Enabling Advanced Net8 Features," in the *Net8 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 a desired feature if application partitioning is used.

Testing Net8 Configuration

To ensure the files are configured correctly:

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

   ```
   SQL> CONNECT internal/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:

   ```
   SQL> UPDATE emp
   set sal = sal + 1000
   where ename = ’miller’;
   commit;
   ```

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

   ```
   SQL> SELECT * from emp;
   ```

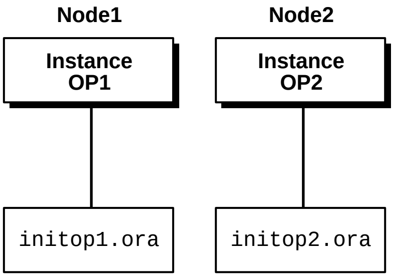
   MILLER’s salary should now be $2,300, indicating that all the instances can see the database.

Understanding the Initialization Parameter Files

An initialization parameter file is an ASCII text file containing a list of parameters.

In Oracle Parallel Server, some initialization parameters must be identical across all instances. Other parameters, however, can have unique values within each instance. Oracle accommodates both common and unique parameter settings by grouping these parameters into two files, the common (`init`*`db_name`*`.ora`) and instance-specific (`init`*`sid`*`.ora` file) parameter files. If you used Oracle Database Configuration Assistant, these file are already established.

Figure 4–1 Instance Initialization Files



Purpose of init*sid*.ora

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

- Unique instance name
- Unique thread number
- Private rollback segments
- Execution of database as an Oracle Parallel Server rather than single instance

The *sid* is the value of the DB_NAME parameter in the `init`*`db_name`*`.ora` file and the thread ID. For instance, if the DB_NAME is `op`, and the first instance has a thread ID of 1, its SID is `op1`; the second instance uses the SID `op2` to identify its instance; and so on.

Example 4–1 and Example 4–2 show the contents of `init*sid*.ora` files for two instances with node numbers of 1 and 2 that Oracle Database Configuration Assistant created:

Example 4–1 initop1.ora

```
ifile='C:\OracleSW\admin\op\pfile\initop.ora'
rollback_segments=(rbs1_1,rbs1_2)
thread=1
parallel_server=true
instance_name=op1
remote_login_passwordfile=exclusive
```

Example 4–2 initop2.ora

```
ifile='C:\OracleSW\admin\op\pfile\initop.ora'
rollback_segments=(rbs2_1,rbs2_2)
thread=2
parallel_server=true
instance_name=op2
remote_login_passwordfile=exclusive
```

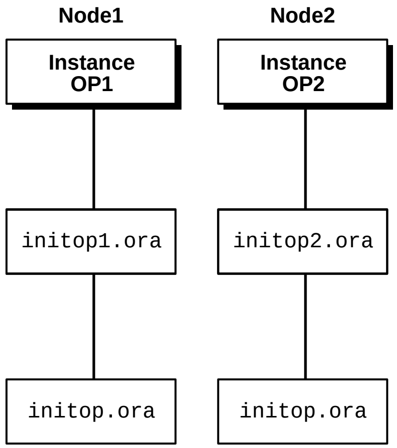
The parameters are described in the following table:

Parameter	**Description**
IFILE	Identifies the path and name of the `init`*`db_name`*`.ora` file to include
ROLLBACK_SEGMENTS	Specifies one or more rollback segments to allocate to this instance
THREAD	Specifies the number of the redo thread that is to be used by the 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. A value of zero causes an available, enabled public thread to be chosen. An instance cannot mount a database if the thread is used by another instance or if the thread is disabled.
INSTANCE_NAME	Identifies the name of instance and is used to uniquely identify a specific instance when multiple instances share common service names
INSTANCE_NUMBER	Maps the instance to one free list group for each database object created with the storage parameter FREELIST GROUPS. Oracle Corporation recommends setting the INSTANCE_NUMBER to the same value as the THREAD parameter. **See Also:** *Oracle8i Parallel Server Administration, Deployment, and Performance*
REMOTE_LOGIN_ PASSWORDFILE	Specifies whether Oracle checks for a password file and how many databases can use the password file. This parameter must be set to EXCLUSIVE. EXCLUSIVE specifies that only one instance can use the password file and that the password file contains names other than SYS and INTERNAL. It allows multiple users (other than INTERNAL and SYS) to start up a database.

Purpose of init*db_name*.ora

The `init`*`db_name`*`.ora` file is called by the individual parameter files through the IFILE parameter setting in `init`*`sid`*`.ora` file.

Figure 4–2 Common Initialization Files



All instances must use the same common file. The instance-specific parameter file is optional. When using the instance-specific parameter file, the IFILE parameter within this file must point to the common file using a complete path name.

Example 4–3 shows a init*db_name*.ora file (initop.ora) created for a Hybrid database through Oracle Database Configuration Assistant:

Example 4–3 initop.ora

```
db_name="op"
db_domain=us.acme.com
service_names=op.us.acme.com
db_files=1024  # INITIAL
control_files=("\\.\op_control1", "\\.\op_control2")
open_cursors=100
db_file_multiblock_read_count=8  # INITIAL
db_block_buffers=13816  # INITIAL
shared_pool_size=19125248  # INITIAL
large_pool_size=18087936
java_pool_size=2097152
log_checkpoint_interval=10000
log_checkpoint_timeout=1800
processes=50  # INITIAL
parallel_max_servers=5  # SMALL
log_buffer=32768  # INITIAL
max_dump_file_size=10240  # limit trace file size to 5M each
global_names=true
oracle_trace_collection_name=""
background_dump_dest=C:\OracleSW\admin\op\bdump
user_dump_dest=C:\OracleSW\admin\op\udump
db_block_size=4096
remote_login_passwordfile=exclusive
os_authent_prefix=""
distributed_transactions=10
mts_dispatchers="(protocol=TCP)(lis=listeners_op)"
compatible=8.1.0
sort_area_size=65536
sort_area_retained_size=65536
```

Take note of the following parameters:

Parameter	Description
BACKGROUND_ DUMP_DEST	Specifies the directory path where debugging trace file for background processes (LGWR, DBWR*n*, and so on) are written during Oracle operations
CONTROL_FILES	Specifies the control files
DB_NAME	Specifies the name of the database, op, entered during installation or database creation

Parameter	Description
DB_DOMAIN	Specifies the database domain, `us.acme.com`, in which the database is located entered during installation or database creation. When possible, Oracle Corporation recommends that your database domain mirror the network domain.

MTS_DISPATCHERS	Enables **multi-threaded server (MTS)** for this database.
	MTS_DISPATCHERS may contain many attributes. At a minimum, Oracle Corporation recommends setting the following attributes:
	■ PROTOCOL (PRO or PROT)
	Specifies the network protocol for which the dispatcher generates a listening endpoint
	■ LISTENER (LIS or LIST)
	Specifies an alias name for the listener(s) with which the **PMON process** registers dispatcher information. The alias should be set to a name which is resolved through a naming method, such a `tnsnames.ora` file
	Oracle Corporation recommends setting MTS_DISPATCHERS as follows:
	`mts_dispatchers="(protocol=tcp)(listener=listeners_`*`db_name`*`)"`
	`listener_`*`db_name`* is resolved through a naming method, such as a `tnsnames.ora` file on the server or an Oracle Names server.
	For example, the MTS_DISPATCHERS parameter can be set as follows in the `init`*`db_name`*`.ora` file:
	`mts_dispatchers="(protocol=tcp)(listener=listeners_op)"`
	`listeners_op` can be then resolved through a local `tnsnames.ora` file as follows:
	`listeners_op=`
	` (description=`
	`  (address_list=`
	`   (address=(protocol=tcp)(host=idops1)(port=1521))`
	`   (address=(protocol= tcp)(host=idops2)(port=1521))))`
	The entry should contain only the listener address, not the service name information in the CONNECT_DATA portion of a connect descriptor.
	See Also:
	■ "Multi-Threaded Server Configuration" on page 3-13
	■ Chapter 2, "Net8 Concepts," and Chapter 9, "Configuring Multi-Threaded Server," in the *Net8 Administrator's Guide* for an overview of MTS

Parameter	Description
SERVICE_NAMES	Specifies the names of the database services on the network, `op.us.acme.com`. By default, Oracle Universal Installer and Oracle Database Configuration Assistant create a service name that includes the entire global database name, a name comprised of the database name (DB_NAME) and domain name (DB_DOMAIN), entered during installation or database creation. It is possible to provide multiple services names (by individual SERVICE_NAMES entries) so that different usages of a instance can be identified separately. Service names can also be used to identify a service that is available from multiple instances through the use of replication.
USER_DUMP_DEST	Specifies the directory path where the server writes debugging trace files on behalf of a user process

See Also: *Oracle8i Reference* for a complete description of theses and other parameters.

Configuring Recovery Manager for Backup and Recovery

This section explains how to configure archive logs to enable you to use Recovery Manager (RMAN) for backup and recovery of an Oracle Parallel Server database.

To configure RMAN for Oracle Parallel Server, perform the following tasks as described in this section:

1. Configure your directories so that all archive log files are accessible by all nodes participating in backup and recovery.
2. Configure the archiver so it can write to multiple destinations.

See Also:

- *Oracle8i Parallel Server Administration, Deployment, and Performance* for general information about backing up Oracle Parallel Server
- *Oracle8i Recovery Manager User's Guide and Reference* for general information about RMAN
- *Oracle Enterprise Manager Administrator's Guide* for information about using Oracle Enterprise Manager for recovery

Configuring Directories for RMAN

To enable RMAN to back up and recover an Oracle Parallel Server database in one step and to use RMAN by way of the Oracle Enterprise Manager Recovery Wizard, all nodes must use the same name for the archive log and must have access to all archive logs. When Oracle generates each archive log, Oracle records the name of the log in the control file or in the recovery catalog. RMAN accesses the archive log files by this name regardless of which node is running RMAN. The easiest way to configure this is to share the archive log directories, as explained under the following headings.

- Configuring Shared Archive Log Destinations on UNIX
- Configuring Shared Archive Log Destinations on Windows NT

Configuring Shared Archive Log Destinations on UNIX

To configure shared archive log destinations on UNIX using NFS (Network File Server), create the same directory structure for the archive logs on every instance. For a three-node cluster, for example, one of the entries is the local archive log destination, and the other two entries are the NFS mounting points for the remote archive logs. Create the following directory structures on each node.

```
$ORACLE_HOME/admin/db_name/arch1
$ORACLE_HOME/admin/db_name/arch2
$ORACLE_HOME/admin/db_name/arch3
```

Each instance writes archive logs to its local archive directory and to the remote directories.

- For the first node, the entry for `arch1` is the local archive log destination for the instance. The other entries for `arch2` and `arch3` are the mount points for the archive logs for the instances on the second and third nodes.
- For the second node, the entry for `arch1` is a mount point for the remote archive log on the first node, the second entry for `arch2` is the local archive log directory, and the third entry for `arch3` is a mount point for the remote archive log on the third node.
- For the third node, the entry for `arch1` is a mount point for the remote archive log on the first node, the second entry for `arch2` is a mount point for the remote archive log on the second node, and the third entry for `arch3` is a mount point for the local archive log directory.

Note: The directories on the remote hosts are mounted by way of NFS.

Special NFS Considerations

Exercise caution when using NFS in Parallel Server environments. If you use "hard NFS" (default), you can block the entire cluster if the remote directories become inaccessible. This might occur as a result of a hardware failure. For this reason, Oracle Corporation strongly recommends that you use NFS implemented for high availability or soft-mounted NFS directories, as explained in the following sections:

- NFS Implemented for High Availability
- Soft-Mounted NFS Directories

NFS Implemented for High Availability The optimal solution is to use a NFS implemented for high availability. This solution uses the exported NFS directory stored on the shared disks of a cluster. One node is the primary node that is used to allow access to the files. If this node fails, a failover process changes the access path to a backup node that also has access to the shared disks. If your hardware supports NFS for high availability, consult your vendor to configure this feature. Otherwise, continue with the procedures under the next heading.

Soft-Mounted NFS Directories Soft mounting means that a process attempting to access the mounted directory is not blocked until the directory becomes available after a failure.

Contact your hardware vendor if your cluster supports soft mounted NFS directories between the nodes in a cluster. Consult your vendor documentation because the commands to configure this are operating system dependent.

On Sun Solaris, for example, create a soft mounted directory using the following commands:

```
mount -F NFS -o soft,rw,retry=10,timeo=30 node1:
/ORACLE_HOME/admin/db_name/arch1
/ORACLE_HOME/admin/db_name/arch1
```

To ensure that each node generates archive logs in its local partition, set the LOG_ARCHIVE_DEST parameter equal to the path for the local archive log file. Using the previous example, make the following entries in the parameter files for the three instances:

In `initop1.ora` enter:

```
log_archive_dest_1="location=/ORACLE_HOME/admin/db_name/arch1"
```

In initop2.ora enter:

```
log_archive_dest_1="location=/ORACLE_HOME/admin/db_name/arch2"
```

In initop3.ora enter:

```
log_archive_dest_1="location=/ORACLE_HOME/admin/db_name/arch3"
```

Oracle Corporation recommends mirroring an additional copy or your archived logs from that node to a remote host. This is explained in "Configuring the Archiver to Write to Multiple Log Destinations" on page 4-14.

Configuring Shared Archive Log Destinations on Windows NT

To configure shared archive logs on Windows NT:

1. Assign an unused drive letter to each node in the cluster. For example, if you have a cluster comprising three nodes named idops1, idops2, and idops3, and if drive letters J, K, and L are unused, assign these letters to the nodes as shown in the following table:

Node Name	Drive Letter
idops1	J:
idops2	K:
idops3	L:

2. Use the Windows NT Disk Administrator application to create new logical partitions containing Windows NT File System (NTFS).

   Each partition will be a local archive log destination for the instance running on that node. To configure this, assign the drive letter owned by that node to the new partition. Continuing with the example in Step 1, on idops1, create a new partition named "J:", on idops2, create a new partition named "K:", and so on. When you create each new partition, also create a directory hierarchy called \archivelogs as shown in the following table:

Node Name	Command
idops1	mkdir J:\archivelogs
idops2	mkdir K:\archivelogs
idops3	mkdir L:\archivelogs

3. On each node, share the new NTFS partition with the other nodes using the following command syntax:

   ```
   net share <db_name>_logs=<drive_letter>:\
   ```

   using the variables *db_name* and *drive_letter* as in the example shown in the following table, where the database name is `op`:

Node Name	Command
`idops1`	`net share op_logs=J:\`
`idops2`	`net share op_logs=K:\`
`idops3`	`net share op_logs=L:\`

4. Use Windows NT Explorer to set permissions on these shared drives.

5. Map the shared drives from the remote nodes in the cluster using the same drive letters with the command:

   ```
   net use \\<node_name>\<db_name>_logs <drive_letter>:
   ```

   For this example, use the variables *node_name*, *db_name*, and *drive_letter* as in the following entries:

   On `idops1`, that has local drive J:, enter

   ```
   net use \\node2\OP_logs K:
   net use \\node3\OP_logs L:
   ```

   On `idops2`, that has local drive K:, enter:

   ```
   net use \\node1\OP_logs J:
   net use \\node3\OP_logs L:
   ```

   On `idops3`, that has local drive L:, enter:

   ```
   net use \\node1\OP_logs J:
   net use \\node2\OP_logs K:
   ```

6. Ensure that each node generates archive logs in its local partition. To do this, set the LOG_ARCHIVE_DEST_*n* parameter in each instance’s parameter file as in the following entries continuing with the example:

   In `initop1.ora` enter:

   ```
   log_archive_dest_1="location=J:\archivelogs"
   ```

In `initop2.ora` enter:

```
log_archive_dest_1="location=K:\archivelogs"
```

In `initop3.ora` enter:

```
log_archive_dest_1="location=L:\archivelogs"
```

Note: You can use the LOG_ARCHIVE_DEST_*n* parameter to configure up to 5 log archive destinations. For more information about this parameter, refer to the *Oracle8i Reference*.

Configuring the Archiver to Write to Multiple Log Destinations

After you have configured your directories, complete the steps described in this section to configure the archiver so it can write to multiple destinations. Multiple archive log destinations avoid single-points-of-failure by making the archive logs for a failed node available to other nodes for recovery processing.

Configure each node to archive to its local disk and to a remote disk. For the remote destination disk, Oracle Corporation recommends that you arrange your nodes in a circular sequence. Do this to allow the first node to write to second node, the second node to write to the third node, and so on. The last node should write to the first node. This way, each node writes to a remote archive log file as well as to a local file.

Configure your archive log destinations for UNIX or Windows, as described in the following sections:

- Configuring Multiple Archive Log Destinations on UNIX
- Configuring Multiple Archive Log Destinations on Windows NT

Configuring Multiple Archive Log Destinations on UNIX

There are two methods for configuring archive log destinations on UNIX as described under the following headings:

- Configuring Shared Archive Log Destinations
- Configuring Non-shared Log Destinations

Configuring Shared Archive Log Destinations To configure multiple destinations on UNIX using shared archive log destinations, add the following initialization parameters to the previous configuration example for UNIX:

In `initop1.ora`

```
log_archive_dest_2="location=/ORACLE_HOME/admin/db_name/arch2"
```

In `initop2.ora`

```
log_archive_dest_2="location=/ORACLE_HOME/admin/db_name/arch3"
```

In `initop3.ora`

```
log_archive_dest_2="location=/ORACLE_HOME/admin/db_name/arch1"
```

Configuring Non-shared Log Destinations If your cluster hardware does not support shared directories with NFS, back up all local files with RMAN. For recovery, copy all the log files to the node from which you want to begin recovery. To automate this, create a shell script to store the necessary remote copy commands. Then to enable RMAN to find the logs, save the logs in a directory hierarchy with the same name as the source directory. On node 1 use the following script:

```
#!/bin/sh
sqlplus system/manager@node1 @switchlog.sql
rcp node2:/ORACLE_HOME/admin/db_name/arch2/*
/ORACLE_HOME/admin/db_name/arch2
rcp node3:/ORACLE_HOME/admin/db_name/arch3/*
/ORACLE_HOME/admin/db_name/arch3
```

The `switchlog.sql` script that is called by the previous script ensures you retrieve all the log files. The contents of `switch.sql` should be:

```
#!/bin/sh
alter system archive log current;
exit
```

Configuring Multiple Archive Log Destinations on Windows NT

For example, add the following initialization parameters to the "Configuring Shared Archive Log Destinations on Windows NT" on page 4-12:

In `initop1.ora`

```
log_archive_dest_2="location=K:archivelogs"
```

In `initop2.ora`

```
log_archive_dest_2="location=L:archivelogs"
```

In `initop3.ora`

```
log_archive_dest_2="location=J:archivelogs"
```

To access remote archive log directories from your database, configure the OracleService*sid* to start with a Windows account that has permission to write to this directory. Otherwise, attempts to do so produce the following message:

```
ORA-9291: sksachk: invalid device specified for archive
destination.
```

Making a Consistent Backup

To perform a closed, consistent backup with Oracle Enterprise Manager's Backup Wizard, you must shut down all instances except the first node's instance.

See Also:

- *Oracle8i Parallel Server Administration, Deployment, and Performance* for general information about backing up Oracle Parallel Server
- *Oracle8i Recovery Manager User's Guide and Reference* for general information about RMAN
- *Oracle Enterprise Manager Administrator's Guide* for information about using Oracle Enterprise Manager for recovery

5

Configuring High-Availability Features

This chapter describe how to configure Oracle Parallel Server high-availability features.

Specific topics discussed are:

- Transparent Application Failover
- Primary and Secondary Instances

Transparent Application Failover

Transparent application failover (TAF) instructs Net8 to fail over an established connection that has failed to a different instance. This enables the user to continue to work using the new connection as if the original connection had never failed.

TAF involves manual configuration of a net service name that includes the FAILOVER_MODE parameter included in the CONNECT_DATA portion of the connect descriptor.

This sections covers the following topics:

- FAILOVER_MODE Parameters
- TAF Implementation
- Verification

FAILOVER_MODE Parameters

The FAILOVER_MODE parameter must be included in the CONNECT_DATA portion of a connect descriptor. FAILOVER_MODE may contain the following parameters:

PARAMETER	Description
BACKUP	Specifies a different net service name for backup connections. A backup should be specified when using PRECONNECT to pre-establish connections.
TYPE **(Required)**	Specifies the type of failover. Three types of Net8 failover functionality are available by default to Oracle Call Interface (OCI) applications: ▪ SESSION: Fails over the session; that is, if a user's connection is lost, a new session is automatically created for the user on the backup. This type of failover does not attempt to recover selects. ▪ SELECT: Enables users with open cursors to continue fetching on them after failure. However, this mode involves overhead on the client side in normal select operations. ▪ NONE: This is the default, in which no failover functionality is used. This can also be explicitly specified to prevent failover from happening.
METHOD	Determines how fast failover occurs from the primary node to the backup node: ▪ BASIC: Establishes connections at failover time. This option requires almost no work on the backup server until failover time. ▪ PRECONNECT: Pre-establishes connections. This provides faster failover but requires that the backup instance be able to support all connections from every supported instance.
RETRIES	Specifies the number of times to attempt to connect. If DELAY is specified, RETRIES defaults to five retry attempts.
DELAY	Specifies the amount of time in seconds to wait between connect attempts. If RETRIES is specified, DELAY defaults to one second

TAF Implementation

Depending on the FAILOVER_MODE parameters, TAF can be implemented in a number of ways. Oracle recommends the following methods:

- Implementing TAF with Connect-Time Failover and Client Load Balancing
- Retrying a Connection
- Pre-Establishing a Connection

Implementing TAF with Connect-Time Failover and Client Load Balancing TAF can be implemented with connect-time failover and client load balancing for multiple addresses. In the following example, Net8 connects randomly to one of the listener addresses on `idops1` or `idops2`. If the instance fails after the connection, Net8 fails over to the other node's instance, reserving any SELECT statements in progress.

```
op.us.acme.com=
 (description=
  (load_balance=on)
  (failover=on)
  (address=
       (protocol=tcp)
       (host=idops1)
       (port=1521))
 (address=
       (protocol=tcp)
       (host=idops2)
       (port=1521))
  (connect_data=
     (service_name=op.us.acme.com)
      (failover_mode=
        (type=select)
        (method=basic))))
```

Retrying a Connection TAF also provides the ability to automatically retry connecting if the first connection attempt fails with the RETRIES and DELAY parameters. In the following example, Net8 tries to connect to the listener on idops1. If the connection attempt fails, Net8 waits 15 seconds before trying to connect again. Net8 attempts to connect up to 20 times

```
op.us.acme.com=
 (description=
  (address=
       (protocol=tcp)
       (host=idops1)
       (port=1521))
  (connect_data=
     (service_name=op.us.acme.com)
      (failover_mode=
        (type=select)
        (method=basic)
        (retries=20)
        (delay=15))))
```

Pre-Establishing a Connection A backup connection can be pre-established. The initial and backup connections must be explicitly specified. In the following example, Net8 connects to the listener on idops1. If idops1 fails after the connection, Net8 fails over to idops2, reserving any SELECT statements in progress.

```
op.acme.com=
 (description=
  (address=
       (protocol=tcp)
       (host=idops1)
       (port=1521))
  (connect_data=
     (service_name=op.us.acme.com)
     (instance_name=op1)
     (failover_mode=
       (backup=op2.acme.com)
       (type=select)
       (method=preconnect))))
```

```
op2.acme.com=
 (description=
  (address=
       (protocol=tcp)
       (host=idops2)
       (port=1521))
  (connect_data=
     (service_name=op.us.acme.com)
     (instance_name=op2)))
```

Verification

You can query FAILOVER_TYPE, FAILOVER_METHOD, and FAILED_OVER columns in the V$SESSION view to verify that TAF is correctly configured.

> **See Also:** *Oracle8i Reference* for more information about the V$SESSION view

Primary and Secondary Instances

Primary and secondary instances specify that one instance accept connections and the other instance only accept connections if the primary instance fails. This feature can only be implemented for a two-instance Oracle Parallel Server.

This section contains the following sections:

- Overview
- Initialization File Configuration
- Client Configuration
- Listener Configuration
- Connecting to Secondary Instances

Overview

An instance is the primary instance when ACTIVE_INSTANCE_COUNT=1 is set in the `init`*`sid`*`.ora` 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 **multi-threaded server (MTS)** is configured 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 which the dispatcher(s) register information. This is resolved to a list of listener address through a **service registration**, such as a `tnsnames.ora` file. This enables the primary instance to accept connections from its local listener, as well as the secondary instance's listener.

A secondary instance also has the ACTIVE_INSTANCE_COUNT=1 setting but does not register with its listener. Therefore, the secondary instance cannot accept client connections through its listener.

If the primary instance fails, the secondary instance assumes the primary role and registers with its listeners. When the failed instance can once again start, it starts up as the secondary instance, and does not register its dedicated and shared servers with its listeners.

Clients connected to the failed primary instance are failed over to the secondary instance if TAF is configured. Clients that connect to the Oracle Parallel Server after the primary instance fails are routed automatically to the secondary instance.

See Also:

- "Transparent Application Failover" on page 5-2 to configured TAF
- See *Oracle8i Parallel Server Concepts* for further general information about primary and secondary instances

Initialization File Configuration

To enable primary and secondary instance configuration, the instance initialization file, `init`*`sid`*`.ora`, must be configured with the ACTIVE_INSTANCE_COUNT parameter for each instance. The value must be 1 on both instances.

```
active_instance_count=1
```

Client Configuration

Oracle recommends configuring a **connect descriptor** on clients that use an address list that contains the listener addresses for the primary instance and the secondary instance.The LOAD_BALANCE parameter must be set to OFF, since 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:

```
op.us.acme.com=
 (description=
  (load_balance=off)
  (address=(protocol=tcp)(host=idops1)(port=1521))
  (address=(protocol=tcp)(host=idops2)(port=1521))
  (connect_data=
     (service_name=op.us.oracle.com)))
```

Oracle does not recommend setting LOAD_BALANCE=ON. If you do, half of the connections try the listener on the secondary instance, which fail to provide a connection. The client then tries the listener on the primary instance, which succeeds. Oracle recommends sending all connections to the active instance first.

See Also:

- Chapter 6, "Configuring Naming Methods", in the *Net8 Administrator's Guide* to configure a connect descriptor
- Chapter 8, "Enabling Advanced Net8 Features", in the *Net8 Administrator's Guide* to configure an address list and multiple address options, including connect-time failover and client load balancing

Listener Configuration

Remove the SID_LIST_*listener_name* information 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 following `listener.ora` file:

Old listener.ora File	Modified listener.ora File
listener=	listener=
(description=	(description=
(address=	(address=
(protocol=tcp)	(protocol=tcp)
(host=idops1)	(host=idops1)
(port=1521)))	(port=1521)))
sid_list_listener=	
(sid_desc=	
(oracle_home=/orahome81)	
(sid_name=op1))	

Note: If you want to connect to a secondary instance do not remove the SID_LIST_*listener_name* entry. Instead, see the next section, "Connecting to Secondary Instances".

Connecting to Secondary Instances

In some situations administrators may wish to connect to the secondary instance even when the primary instance is alive. For example, the administrator may want perform some batch operation on the database.

Remote Login

Administrators can connect the secondary instance by logging into the machine and performing the batch operation.

Connecting from a Remote Client

Connecting to the secondary instance from a remote client involves some configuration that is dependent on whether or not the operational clients—that is, clients that are not performing administrative activities—are configured with connect descriptors that use SERVICE_NAMES or SIDs, as described in the following sections:

- Clients that Use Connect Descriptor with SERVICE_NAMES
- Clients that Use Connect Descriptor with SID

Clients that Use Connect Descriptor with SERVICE_NAMES Administrators that have clients that use the SERVICE_NAME parameter in their connect descriptors can connect to the secondary instance with one of the following method:

- Create a connect descriptor for each instance that uses the SID of the instance in the CONNECT_DATA portion of the connect descriptor

  In the following example, the client can connect to `idops1` and `idops2` using `op1.us.oracle.com` and `op2.us.oracle.com`.

  ```
  op1.us.acme.com=
   (description=
    (address=(protocol=tcp)(host=idops1)(port=1521))
    (connect_data=
       (service_name=op1.us.oracle.com)))
  op2.us.acme.com=
   (description=
    (address=(protocol=tcp)(host=idops1)(port=1521))
    (connect_data=
       (service_name=op2.us.oracle.com)))
  ```

  No further configuration is required.

- Set or modify the GLOBAL_DBNAME parameter in the `listener.ora` file to a name that administrators alone can use and specify the name as the value for the SERVICE_NAME parameter in the CONNECT_DATA portion of the connect descriptor.

The installation and configuration process should have created a default `listener.ora` files without the GLOBAL_DBNAME parameter. You system may have `listener.ora` files that contain the GLOBAL_DBNAME parameter. The GLOBAL_DBNAME parameter specifies a value that typically matches the SERVICE_NAMES parameter in the `init`*`db_name`*`.ora` file. In the following example, the SID_LIST_*listener_name* information specifies an instance named op1 and a database named op.us.acme.com.

```
listener=
 (description=
  (address=
   (protocol=tcp)
   (host=idops1)

   (port=1521)))
sid_list_listener=
 (sid_desc=
   (oracle_home=/orahome81)
   (global_dbname=ops.us.acme.com)
   (sid_name=op1))
```

The SID_LIST_*listener_name* static information is not used, because the instance (if primary) has already registered this information with the listener through dynamic service registration, or the instance (if secondary) does not register with its local listener.

Since the service name is already registered with the listener, the value of the GLOBAL_DBNAME parameter, if present, is not used. Therefore, you can set the parameter to another value that is different from the registered service name.

If the GLOBAL_DBNAME is not present, add it and set it to a value that is different from the registered service name; if the GLOBAL_DBNAME is present, replace the GLOBAL_DBNAME value with different name than the service name.

This new static information does not match the information dynamically registered with the listener through the service registration. Therefore, operational clients are directed to the primary instance and administrative clients that specify the modified service name in the connect descriptor can connect to the secondary instance.

For example, the same `listener.ora` can modified with a global database name of `adminop.us.acme.com`.

Old listener.ora File	Modified listener.ora File
`listener=`	`listener=`
` (description=`	` (description=`
`  (address=`	`  (address=`
`   (protocol=tcp)`	`   (protocol=tcp)`
`   (host=idops1)`	`   (host=idops1)`
`   (port=1521)))`	`   (port=1521)))`
`sid_list_listener=`	`sid_list_listener=`
` (sid_desc=`	` (sid_desc=`
`   (global_dbname=op.us.acme.com)`	**`   (global_dbname=adminop1.us.acme.com)`**
`   (oracle_home=/orahome81)`	`   (oracle_home=/orahome81)`
`   (sid_name=op1))`	`   (sid_name=op1))`

The `listener.ora` file on the other node would also be modified in the following manner:

Old listener.ora File	Modified listener.ora File
`listener=`	`listener=`
` (description=`	` (description=`
`  (address=`	`  (address=`
`   (protocol=tcp)`	`   (protocol=tcp)`
`   (host=idops2)`	`   (host=idops2)`
`   (port=1521)))`	`   (port=1521)))`
`sid_list_listener=`	`sid_list_listener=`
` (sid_desc=`	` (sid_desc=`
`   (global_dbname=op.us.acme.com)`	**`   (global_dbname=adminop2.us.acme.com)`**
`   (oracle_home=/orahome81)`	`   (oracle_home=/orahome81)`
`   (sid_name=op2))`	`   (sid_name=op2))`

See Also: Chapter 7, "Configuring the Listener," in the *Net8 Administrator’s Guide* for listener configuration instructions

For those clients that are to perform remote administration, create separate connect descriptors for each of the instances. Ensure that the CONNECT_DATA portion uses a SERVICE_NAME that has a value that matches the new value for the `listener.ora` file's GLOBAL_DBNAME parameter.

In the following example, the client can connect to `idops1` and `idops2` using `adminop1.us.oracle.com` and `adminop2.us.oracle.com`.

```
adminop1.us.acme.com=
 (description=
  (address=(protocol=tcp)(host=idops1)(port=1521))
  (connect_data=
     (service_name=adminop1.us.oracle.com)))
adminop2.us.acme.com=
 (description=
  (address=(protocol=tcp)(host=idops1)(port=1521))
  (connect_data=
     (service_name=adminop2.us.oracle.com)))
```

See Also:

- Chapter 6, "Configuring Naming Methods,"in the *Net8 Administrator's Guide* to configure a connect descriptor
- Chapter 7 "Configuring the Listener," in the *Net8 Administrator's Guide* to configure static service information

Clients that Use Connect Descriptor with SID Administrators that have clients that use the SID parameter in their connect descriptors can connect to the secondary instance by:

1. Configuring an additional administrative listener in the `listener.ora` file.
2. Specifying the appropriate SID in the CONNECT_DATA portion of the connect descriptor.

Both instances should be configured with two listeners:

- One listener, typically with a default name of `listener`, is used for client connections.

  This listener should not contain a SID_LIST_*listener_name* entry in the `listener.ora` file. This listener relies solely on dynamic service registration to obtain information about the database.

- A second administrative listener, with a different name, is used for remote administration.

  Only clients performing remote administration use this listener. This listener does not rely on service registration for database registration information. Instead, it relies on the SID_LIST_*listener_name* entry in the `listener.ora` file.

The installation and configuration process should have already created `listener.ora` files on both nodes with the SID_LIST_*listener_name* information.

To create the administrative listener:

1. Create another listener with a distinctive name, such as `listener_admin`.
2. Give it a port number that is not 1521, the default port number.
3. Replace *listener_name* in SID_LIST_*listener_name* in the `listener.ora` file with the name of the administrative listener.

For example, the following `listener.ora` file has an entry for an administrative listener called `listener_admin` that listens on port 1480 for requests to an instance named `op1` that services database `op.us.acme.com`:

Old listener.ora File	Modified listener.ora File
`listener=`	`listener=`
` (description=`	` (description=`
`  (address=`	`  (address=`
`   (protocol=tcp)`	`   (protocol=tcp)`
`   (host=idops1)`	`   (host=idops1)`
`   (port=1521)))`	`   (port=1521)))`
`sid_list_listener=`	**`listener_admin=`**
` (sid_desc=`	**` (description=`**
`   (oracle_home=/orahome81)`	**`  (address=`**
`   (sid_name=op1))`	**`   (protocol=tcp)`**
	`   (host=idops1)`
	`   (port=1480)))`
	`sid_list_listener_admin=`
	` (sid_desc=`
	`   (global_dbname=op.us.acme.com)`
	`   (oracle_home=/orahome81)`
	`   (sid_name=op1))`

The `listener.ora` file on the other node would also be modified in the following manner:

Old listener.ora File	Modified listener.ora File
listener= (description= (address= (protocol=tcp) (host=idops2) (port=1521))) sid_list_listener= (sid_desc= (global_dbname=op.us.acme.com) (oracle_home=/orahome81) (sid_name=op2))	listener= (description= (address= (protocol=tcp) (host=idops2) (port=1521))) **listener_admin=** **(description=** **(address=** **(protocol=tcp)** **(host=idops2)** **(port=1480)))** **sid_list_listener_admin=** **(sid_desc=** **(global_dbname=op.us.acme.com)** **(oracle_home=/orahome81)** **(sid_name=op2))**

> **See Also:** Chapter 7, "Configuring the Listener," in the *Net8 Administrator's Guide* for listener configuration instructions

4. For those clients that are to perform remote administration, create a separate connect descriptor for each of the administrative listeners.

   For example:

```
op1_admin1=
 (description=
   (address=(protocol=tcp)(host=idops1)(port=1480))
  (connect_data=
     (sid=op1)))
op2_admin2=
 (description=
   (address=(protocol=tcp)(host=idops2)(port=1480))
  (connect_data=
     (sid=op2)))
```

> **See Also:** See Chapter 6, "Configuring Naming Methods," in the *Net8 Administrator's Guide* to configure a connect descriptor.

Part III

Installing and Using Oracle Parallel Server Management

Part III discusses how to install and use Oracle Parallel Server Management. The chapters in Part III are:

- Chapter 6, "Installing and Configuring Oracle Parallel Server Management"
- Chapter 7, "Administering Oracle Parallel Server with Oracle Parallel Server Management"
- Chapter 8, "Monitoring Performance with Oracle Performance Manager"

6

Installing and Configuring Oracle Parallel Server Management

Oracle Enterpise Manager supports administration of Oracle Parallel Server databases with Oracle Parallel Server Management. Oracle Parallel Server Management enables certain parallel components of Oracle Enterprise Manager. A performance monitoring tool called Oracle Performance Manager further enhances Oracle Parallel Server Management, enabling end users to monitor the global V$ view tables for performance.

This chapter describes how to install and configure Oracle Parallel Server Management.

Specific topics discussed are:

- Oracle Parallel Server Management Architecture
- Oracle Parallel Server Management Requirements
- Understanding Oracle Enterprise Manager Setup
- Installing Oracle Enterprise Manager
- Configuring Oracle Enterprise Manager
- Configuring Oracle Performance Manager
- Additional Notes for DBA Studio

Oracle Parallel Server Management Architecture

You can control the activity of Oracle Parallel Servers and their instances using Oracle Parallel Server Management. Oracle Parallel Server Management is a comprehensive, integrated system management solution for an Oracle Parallel Server. Oracle Parallel Server Management enables you to manage multi-instance databases running in heterogeneous environments through an open client-server architecture.

In addition to managing parallel databases, Oracle Parallel Server Management enables you to schedule jobs, perform event management, monitor performance, and obtain statistics to tune parallel databases.

Oracle Enterprise Manager provides database administrators (DBAs) with a powerful set of tools to manage, monitor, and administer even the most complex network of databases from a single workstation, called the Enterprise Manager Console.

> **See Also:** *Oracle8i Parallel Server Administration, Deployment, and Performance* for further information about using views not available with Oracle Enterprise Manager, and for more detail on monitoring and tuning Oracle Parallel Server

Oracle Parallel Server Management within Oracle Enterprise Manager context, contains the components shown in the table below:

Component	Description
Console	The **Console** is a graphical user interface with menus, toolbars, and launch palettes to enable access to Oracle tools. The Console is run from a client machine or a browser. The Console has four windows: ■ Navigator window — Contains an object browser that provides an organized, consistent and hierarchical view of the database objects in the network ■ Group window — Provides a customized, graphical representation of key objects, created by the administrator ■ Event Management window — Enables the administrator to remotely monitor critical database and system events ■ Job window — Provides the administrator with the ability to automate repetitive activities

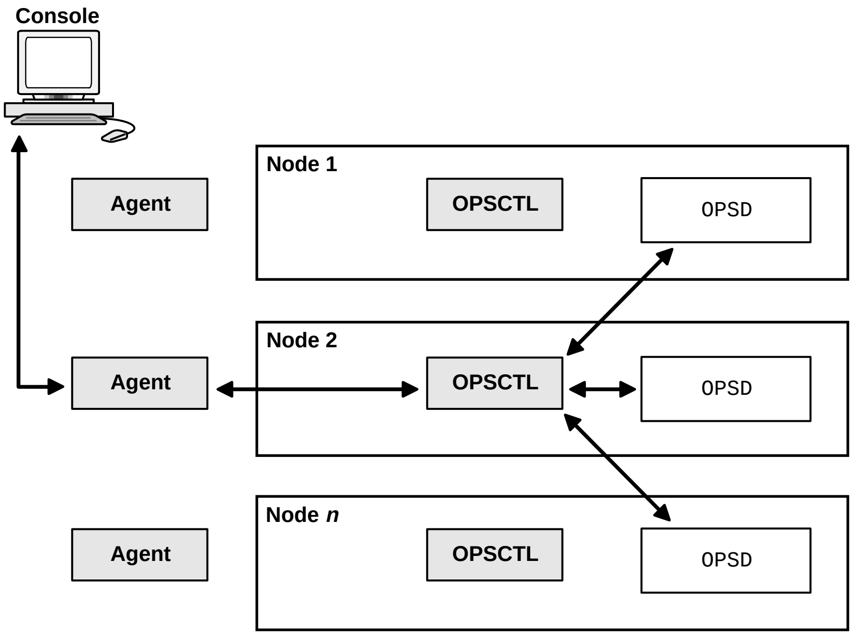
Component	Description
Management Server	The **Management Server** performs functions requested by the Console. The Management Server processes all system management tasks and administers the distribution of these tasks to **Oracle Intelligent Agents** on the managed nodes.
Repository Database	All administrators share a **repository database** which is a set of database tables. The repository contains the accounts of every administrator. The Management Server uses the repository to store all system data, application data, and the state of managed nodes throughout the environment. The repository can be shared by multiple administrators. The repository *must* be on a separate machine from the nodes.
Oracle Intelligent Agents	Oracle Intelligent Agents manages and completes tasks requested from the Console. Once installed, Oracle Intelligent Agent: ■ Listens for and responds to job requests initiated by the Console. ■ Schedules job requests. Requests can include detecting and correcting abnormal conditions, performing standard database administration procedures, and monitoring events. Oracle Intelligent Agent is installed on the nodes.
(Optional) Oracle Performance Manager	**Oracle Performance Manager** enables you to choose from a variety of tabular and graphic performance statistics for Oracle Parallel Servers. The statistics represent the aggregate performance for all instances running on an Oracle Parallel Server. The statistics are displayed in individual charts and include information about data block pings, lock activity, file I/O, and session and user information.
(Optional) Oracle Data Gatherer	**Oracle Data Gatherer** gathers performance statistics for Oracle Performance Manager. Oracle Data Gatherer is installed on each node with Oracle Intelligent Agent.

Oracle Parallel Server Management uses the utilities shown in the table below to manage instances:

Component	Description
OPS Control (OPSCTL)	OPSCTL serves as a single point of control between Oracle Intelligent Agent and the nodes Only one node's Oracle Intelligent Agent is used to communicate to OPSCTL. OPSCTL on that node then communicates to the other nodes through Net8. OPSCTL is installed on the nodes.
OPS Communication Daemon (OPSD)	OPSD receives requests from OPSCTL 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 OPSCTL. OPSD is installed on the nodes. OPSD is only implemented on UNIX operating systems.

On UNIX, Oracle Intelligent Agent invokes OPSCTL to executes jobs. OPSD then receives requests from OPSCTL, as shown in Figure 6–1:

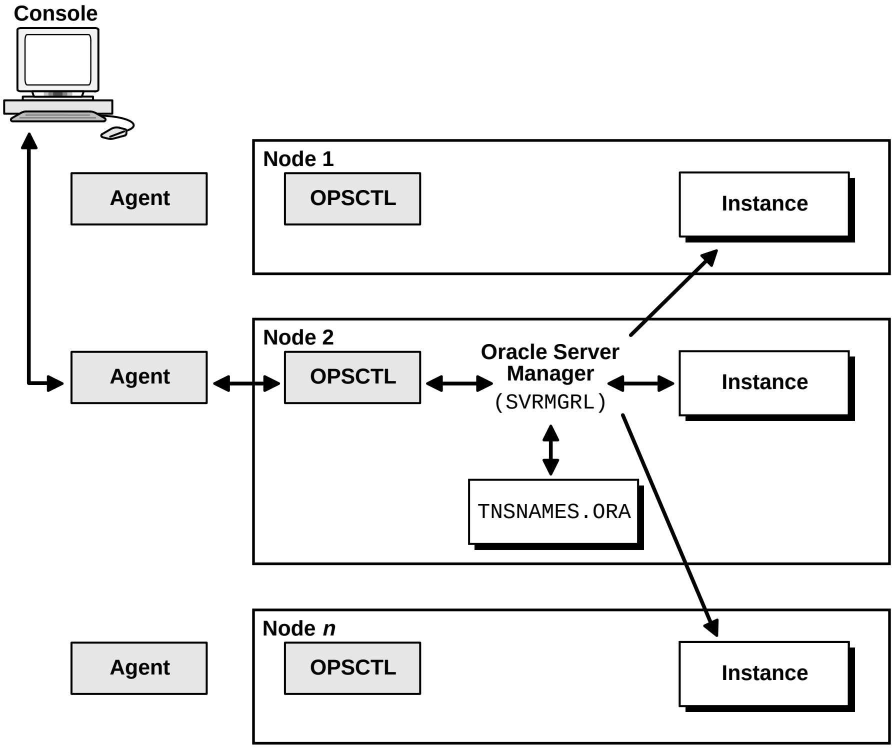
Figure 6–1 OPSCTL Architecture for UNIX



On Windows NT, Oracle Intelligent Agent invokes OPSCTL to executes jobs. OPSCTL then sends requests to Oracle Server Manager. Oracle Server Manager uses information from configured `tnsnames.ora` files on the nodes to connect to a listener which sends the OPSCTL requested action to the instance. Therefore, it is essential to have the `tnsnames.ora` file configured on all nodes in the cluster.

Figure 6–2 shows how OPSCTL operates between the agent and the nodes on Windows NT:

Figure 6–2 OPSCTL Architecture for Windows NT



Oracle Parallel Server Management Requirements

The requirements for each node as shown in the table below must be met to use Oracle Parallel Server Management from the Console. These requirements should have been met if you followed the procedures in Chapter 3, "Installing and Creating an Oracle Parallel Server Database".

Node Requirements	For additional information, see:
Ensure Oracle8*i* Enterprise Edition, Oracle Parallel Server Option, and Oracle Intelligent Agent are installed on each node in the cluster.	Chapter 3, "Installing and Creating an Oracle Parallel Server Database"
Each instance's Oracle System Identifiers (SIDs) in the Oracle Parallel Server must be unique.	"Unique Identification of Instances with the initsid.ora File" on page 1-15
Each node must have a `tnsnames.ora` file configured with a net service name entry for: ▪ The database ▪ Each instance	"Net Service Names (tnsnames.ora file)" on page 3-16
The `listener.ora` file on each node must be configured with: ▪ The listener name ▪ A TCP/IP address for Oracle Enterprise Manager ▪ An entry for the SID of the node in the SID_LIST_*listener_name* section.	"Listener (listener.ora file)" on page 3-14
The `oratab` file and *`db_name`*`.conf` file must be configured on UNIX for OPSCTL. The HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\OSD\PM must be set up on Windows NT for discovery work	▪ "Operating-System Specific Configuration" on page 3-9 if you used Oracle Database Configuration Assistant ▪ "Task 3: Set operating-system specific configuration." on page 3-32 if you used manual methods to create the database
For UNIX, ensure OPSD has been started on each managed node by the *oracle* account at system startup. OPSD resides in `$ORACLE_HOME/bin`.	

Understanding Oracle Enterprise Manager Setup

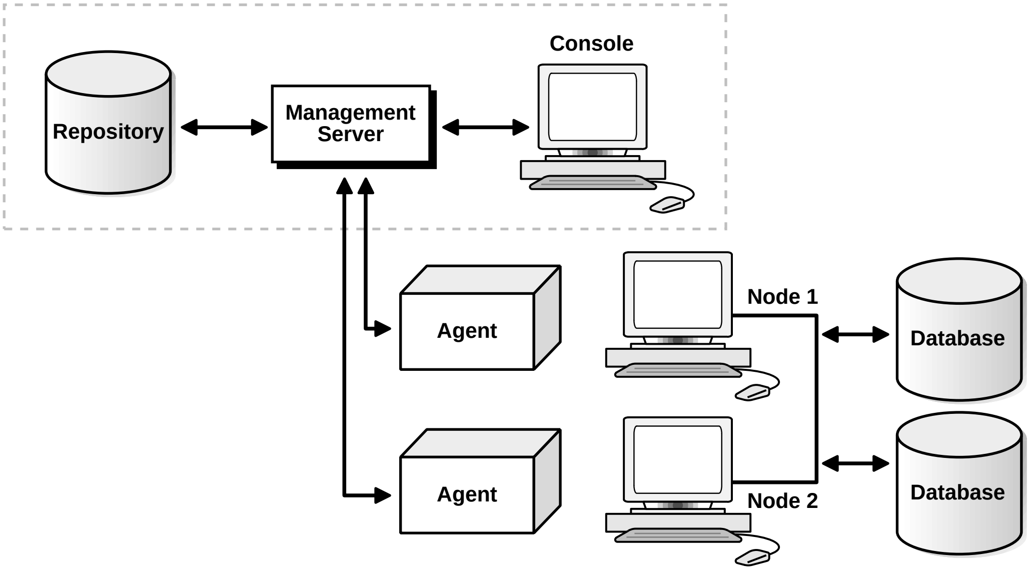
You can run the individual Oracle Enterprise Manager components on separate machines or combine different components on separate machines to collaboratively manage the complete Oracle environment. Two environments are shown below:

Environment 1: Console, Management Server and Repository on Same Machine

The Console, with the aid of the Management Server, remotely manages the databases for both nodes. The Console and the Management Server are running on a Windows NT or Solaris machine with an Oracle8*i* database installed that is only used as a repository. Oracle Intelligent Agent does *not* need to be running.

The nodes share an Oracle8*i* database. The repository is *not* created in this database, and Oracle Intelligent Agent is running on *both* nodes.

Figure 6–3 Console, Management Server and Repository on the Same Machine



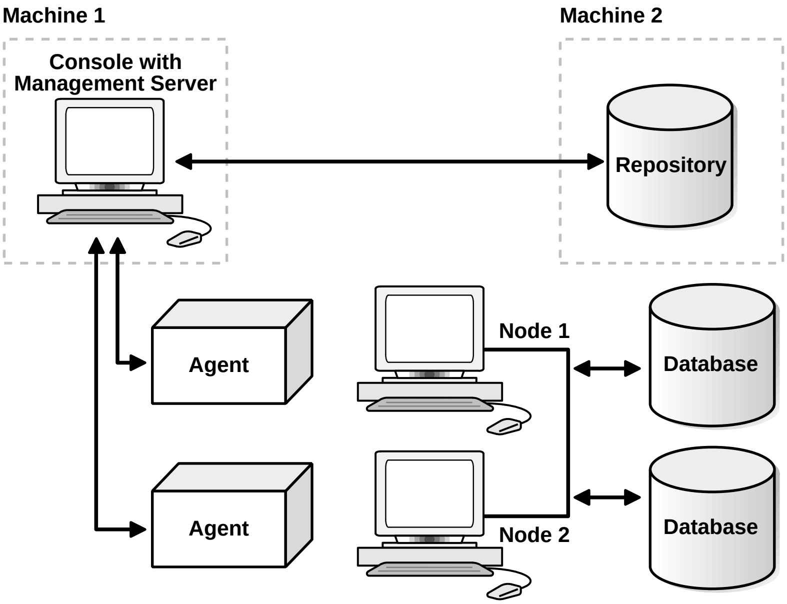
Environment 2: Console, Management Server and Repository on Separate Machines

The Repository is on a separate Windows NT or Solaris machine with an Oracle8*i* database installed that is only being used as a repository. Oracle Intelligent Agent does *not* need to be running.

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

The nodes run an Oracle8*i* database. The repository is *not* created in this database, and Oracle Intelligent Agent is running on *both* nodes.

Figure 6–4 Console, Management Server and Repository on the Separate Machines



Installing Oracle Enterprise Manager

To install Oracle Enterprise Manager:

1. Install the repository database on a machine other than one of the Oracle Parallel Server nodes.

   See Also:

   - *Oracle8i Installation Guide* for Sun Solaris, HP 9000 and AIX-based systems
   - *Oracle8i Installation Guide for Windows NT*

2. Install Oracle Enterprise Manager.

   You can install the Management Server and the Console on the machine where the repository database resides or on individual machines. If you purchased Oracle Diagnostics Pack, you can install Oracle Diagnostics Pack and Oracle Enterprise Manager together.

   See Also:

   - *Oracle8i Installation Guide* for Sun Solaris, HP 9000 and AIX-based systems or *Oracle8i Installation Guide for Windows NT* to install Oracle Enterprise Manager without Oracle Diagnostics Pack
   - *Oracle Diagnostics Pack Installation* to install Oracle Enterprise Manager and Oracle Diagnostics Pack

Note: Oracle Performance Manager can be run in addition to or without Oracle Enterprise Manager. If you choose to run this product as a stand-alone product, Oracle Enterprise Manager does not have to be configured.

Configuring Oracle Enterprise Manager

To configure Oracle Enterprise Manager, perform these tasks:

- Task 1: Configure Repository
- Task 2: Start Oracle Enterprise Manager Components
- Task 3: Discover Nodes
- Task 4: Specify Preferred Credentials for Nodes and Oracle Parallel Server Database

Task 1: Configure Repository

Use Configuration Assistant to create and load your version 2 repository. The repository is a set of tables in an Oracle database which stores data required by Oracle Enterprise Manager. See *Oracle Enterprise Manager Configuration Guide* for complete configuration instructions.

Task 2: Start Oracle Enterprise Manager Components

To use the Console, start the following components:

- Oracle Intelligent Agent
- Management Server
- Console

Oracle Intelligent Agent

Start the Oracle Intelligent Agent on each of the nodes.

At an operating system prompt, enter the command:

```
lsnrctl dbsnmp_start
```

On Windows NT, you can also use the Control Panel:

1. Double-click the Services icon in the Control Panel window.
2. Select the Oracle*HOME_NAME*Agent service.
3. Click Start to start the service.

> **Note:** Additional Oracle Intelligent Agent configuration is required if the ORACLE_HOME is shared, as described in "Oracle Intelligent Agent Configuration" on page B-10.

Management Server

Note: Your Management Server service is started automatically during repository creation if you launched the Configuration Assistant from the Oracle Enterprise Manager installation.

To start the Management Server, at an operating system prompt, enter:

```
oemctrl start oms
```

The command string is case-sensitive and you must enter it with lowercase characters.

To start the Management Server in the background on Solaris, enter:

```
% oemctrl start oms&
```

On Windows NT, you can also use the Control Panel:

1. Double-click the Services icon in the Control Panel window.
2. Select the Oracle*HOME_NAME*ManagementServer service.
3. Click Start to start the service.

The Management Server looks for the OMSCONFIG.PROPERTIES file in the `$ORACLE_HOME/sysman/config` directory on UNIX and *`ORACLE_HOME`*`\sysman\config` directory on Windows operating systems. It contains the connect information required by the Management Server to start up.

Note: Always use Oracle Enterprise Manager Configuration Assistant to create a repository or to change the repository's connect information. The information is encrypted in the OMSCONFIG.PROPERTIES file.

See Also: *Oracle Enterprise Manager Configuration Guide* for optional Console configuration detailed information about editing the OMSCONFIG.PROPERTIES file.

Console

The 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.

To start the Enterprise Manager Console:

1. At an operating system prompt, enter:

   ```
   oemapp console
   ```

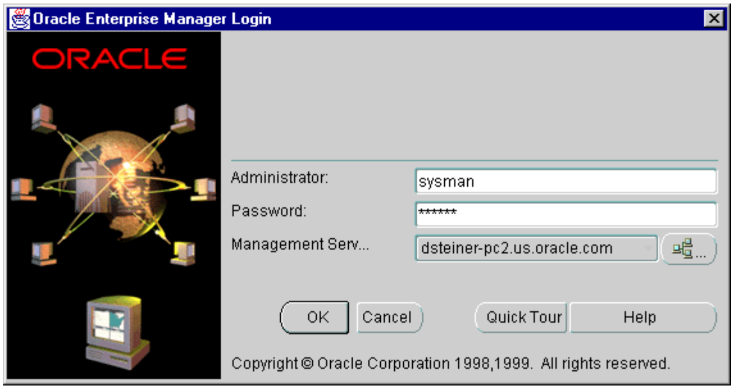
   The command string is case-sensitive and you must enter it with lowercase characters.

   On Windows NT, you can also choose Start > Programs > Oracle - *HOME_NAME* > Oracle Enterprise Management > Oracle Enterprise Manager Console.

2. If you are logging in to the Console for the first time, after the first login dialog appears, enter the default credentials for the super administrator account:

   Administrator = `sysman`

   Password = `oem_temp`



Oracle Enterprise Manager Login

Administrator: sysman

Password: ******

Management Serv... dsteiner-pc2.us.oracle.com

Copyright © Oracle Corporation 1998,1999. All rights reserved.

3. Select a Management Server that is configured with the repository you want to access from the Management Server list.

If the name of the Management Server you want to use does not appear, perform these procedures:

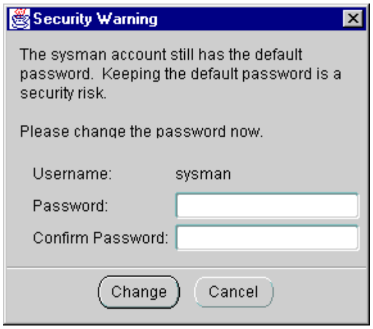
a. Click the Management Servers button. The Management Servers dialog appears.

b. Click the Add button. The Add Management Server dialog appears.

c. Enter in the name of the Management Server, then click OK.

d. Select the Management Server from the list.

A Security Warning dialog appears in which you can change your password.



Security Warning

The sysman account still has the default password. Keeping the default password is a security risk.

Please change the password now.

Username: sysman

Password:

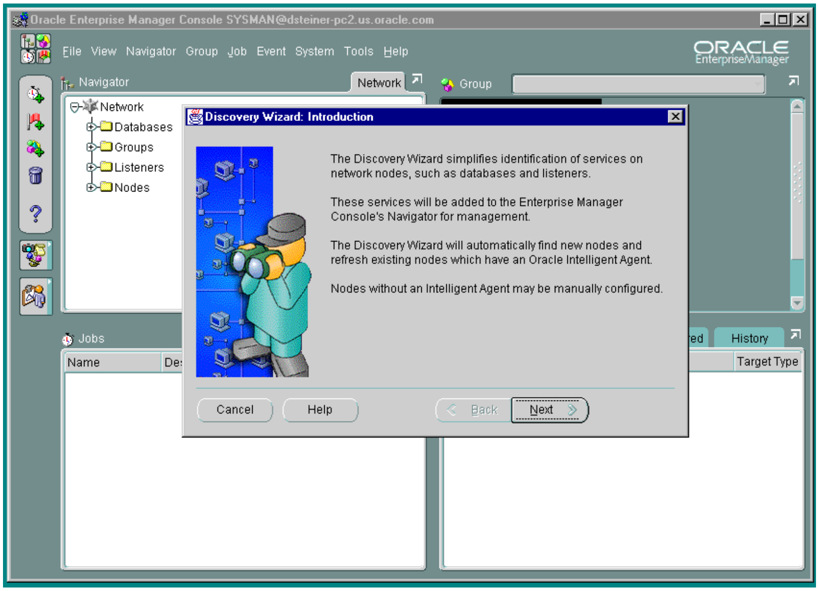
Confirm Password:

4. Enter a password and confirm it.

   The new password you enter is the password you use to log in to the Console for subsequent logins.

Note: Passwords must have no more than 8 characters.

The Console appears with the Discovery Wizard:



The Discovery Wizard simplifies identification of services on network nodes, such as databases and listeners.

These services will be added to the Enterprise Manager Console's Navigator for management.

The Discovery Wizard will automatically find new nodes and refresh existing nodes which have an Oracle Intelligent Agent.

Nodes without an Intelligent Agent may be manually configured.

The first time you start the Console, you must log in as the super administrator. After other administrator accounts have been created using the super administrator account, you can log in as an administrator.

5. Continue with "Task 3: Discover Nodes" on page 6-15.

Note: The Console may also be run from a web browser, allowing administrators to run the Console as a "thin" or "fat" client.

Thin clients use a web browser to connect to a server where Console files are installed, whereas fat clients have Console files installed locally. Run from a web browser, the Console enables you to perform all the same administration tasks you would normally perform from the installed Console, including access to the DBA Management Pack applications. You can perform these tasks on any machine that supports a web browser.

For further information about running the Oracle Enterprise Manager from a web browser, follow the instructions in the *Oracle Enterprise Manager Configuration Guide*

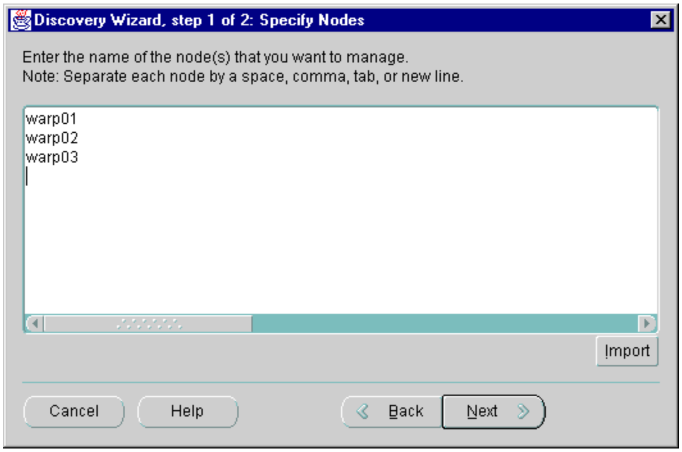
Task 3: Discover Nodes

After the Console has been started, discover network services, such as databases, listeners, and nodes, to populate the Navigator tree.

To discover services:

1. Click Next in the Discovery Wizard: Introduction page.

   The Specify Nodes page appears:



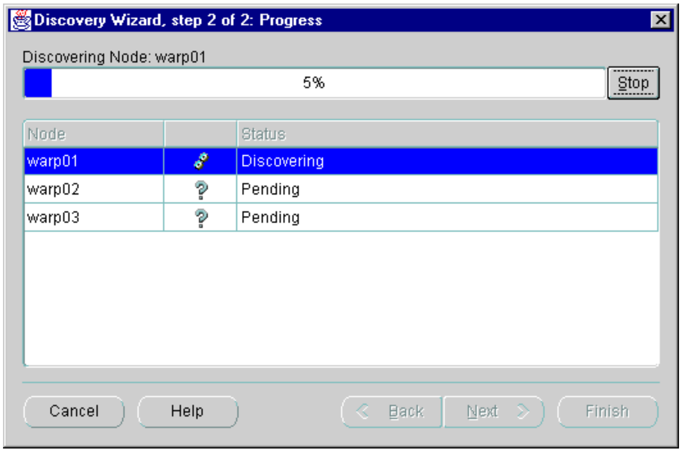
Discovery Wizard, step 1 of 2: Specify Nodes

Enter the name of the node(s) that you want to manage.
Note: Separate each node by a space, comma, tab, or new line.

```
warp01
warp02
warp03
```

2. Enter the host names of Oracle Parallel Server nodes.

The Progress page appears with the discovery status:

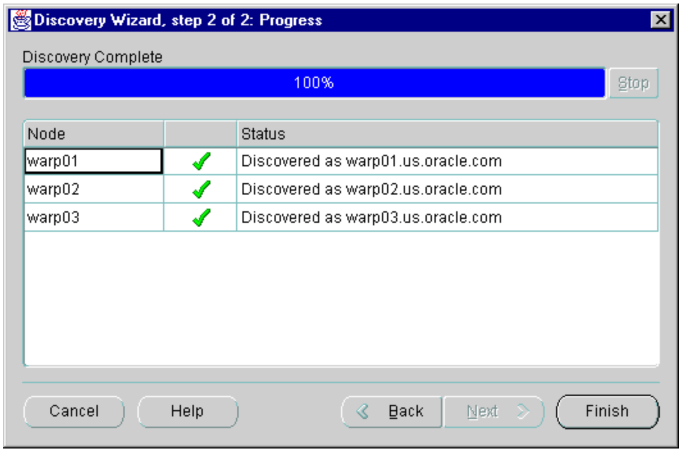


Discovering Node: warp01

5%

Node		Status
warp01		Discovering
warp02	?	Pending
warp03	?	Pending

Successful discovery results in the following:



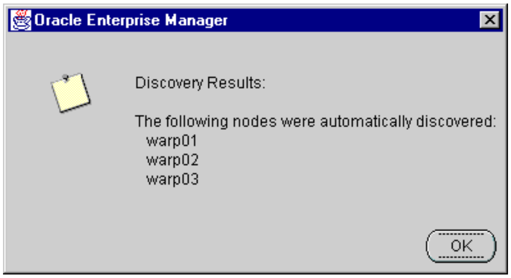
Discovery Complete

100%

Node		Status
warp01	✓	Discovered as warp01.us.oracle.com
warp02	✓	Discovered as warp02.us.oracle.com
warp03	✓	Discovered as warp03.us.oracle.com

3. Click Finish.

The following confirmation dialog box appears:



Discovery Results:

The following nodes were automatically discovered:
warp01
warp02
warp03

Note: An unsuccessful discovery is usually the result of the Oracle Intelligent Agent not being started on the node. See "Resolving Service Discovery Failures" on page C-2.

---

4. Click OK to acknowledge the status dialog.

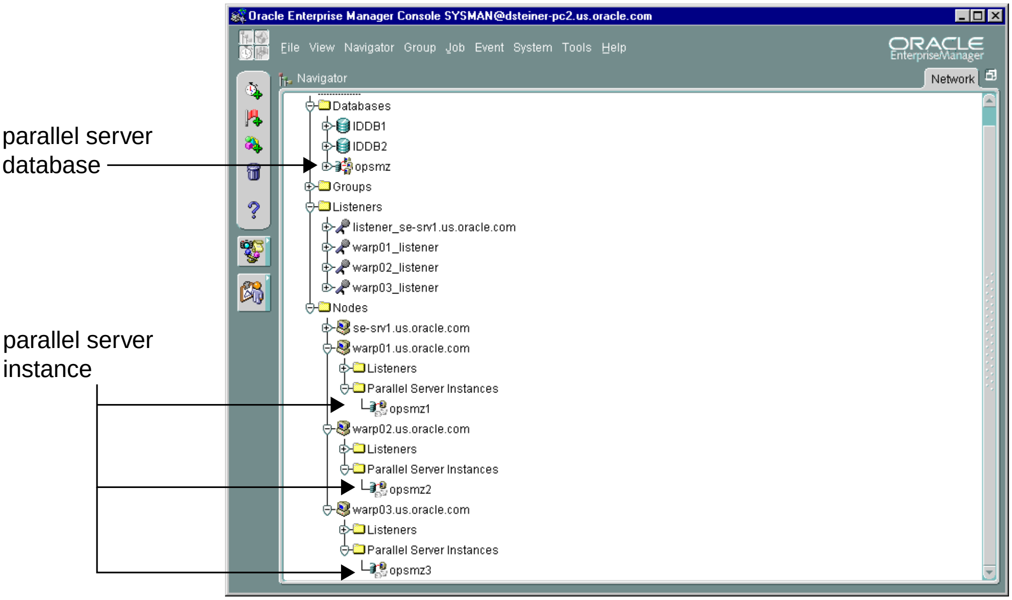
   Once a node is identified, Oracle Enterprise Manager automatically discovers (locates) all the databases and other services running on the managed nodes. When a node is discovered, the services located are:

   - The database it supports
   - Its instances
   - Its listeners

   When an Oracle Parallel Server instance is discovered, a new folder named Parallel Server Instances is created for the node it belongs to.

   By expanding and contracting the folders in the Navigator window, you can see the object discovered.

The following graphic shows an Oracle Parallel Server database named `opsmz` with instance `opsmz1` on host `warp01.us.oracle.com`, instance `opsmz2` on host `warp02.us.oracle.com`, and instance `opsmz3` on host `warp03.us.oracle.com`:


parallel server database
parallel server instance
Oracle Enterprise Manager Console SYSMAN@dsteiner-pc2.us.oracle.com
File View Navigator Group Job Event System Tools Help
ORACLE EnterpriseManager
Navigator
Network

- Databases
  - IDDB1
  - IDDB2
  - opsmz
- Groups
- Listeners
  - listener_se-srv1.us.oracle.com
  - warp01_listener
  - warp02_listener
  - warp03_listener
- Nodes
  - se-srv1.us.oracle.com
  - warp01.us.oracle.com
    - Listeners
    - Parallel Server Instances
      - opsmz1
  - warp02.us.oracle.com
    - Listeners
    - Parallel Server Instances
      - opsmz2
  - warp03.us.oracle.com
    - Listeners
    - Parallel Server Instances
      - opsmz3

Task 4: Specify Preferred Credentials for Nodes and Oracle Parallel Server 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 Oracle Parallel Server, such as when the Oracle Parallel Server 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 Oracle Parallel Server database on the target node. This enables you to expand the Oracle Parallel Server 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.

Likewise, you must 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.

This section covers the following topics:

- Create an Operating System Account
- Grant SYSDBA or SYSOPER Privileges to a User
- Set User Credentials in the Console

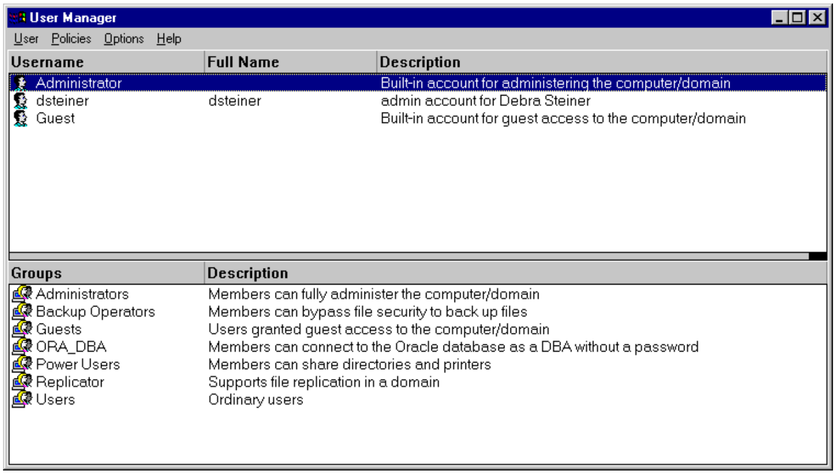
Create an Operating System Account

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

On Windows NT, you must create a Windows NT user account using the next procedure. To create a user account on each node of the cluster:

1. Choose Start > Programs > Administrative Tools (Common) > User Manager:

   The User Manager window appears:

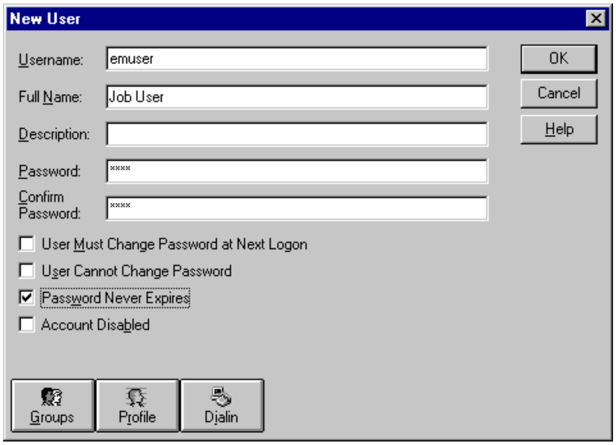


Username	Full Name	Description
Administrator		Built-in account for administering the computer/domain
dsteiner	dsteiner	admin account for Debra Steiner
Guest		Built-in account for guest access to the computer/domain

Groups	Description
Administrators	Members can fully administer the computer/domain
Backup Operators	Members can bypass file security to back up files
Guests	Users granted guest access to the computer/domain
ORA_DBA	Members can connect to the Oracle database as a DBA without a password
Power Users	Members can share directories and printers
Replicator	Supports file replication in a domain
Users	Ordinary users

2. Choose User > New User.

The New User dialog box appears:



New User

Username: emuser

Full Name: Job User

Description:

Password: ****

Confirm Password: ****

□ User Must Change Password at Next Logon

□ User Cannot Change Password

☑ Password Never Expires

□ Account Disabled

3. Enter the appropriate information in the dialog box:

   a. Enter a user name that is up to eight characters in length.

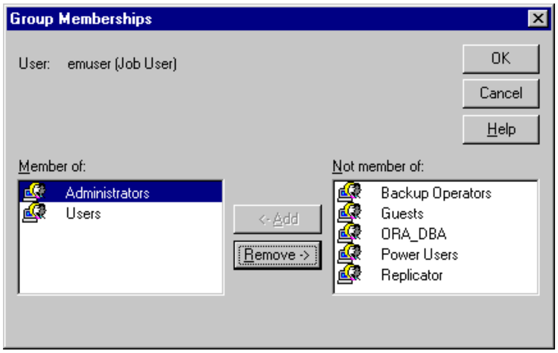
   b. Optionally, enter a full name and description for the user.

   c. Enter and confirm a password that is up to eight characters in length.

   d. Make sure the "User Must Change Password at Next Logon" check box is not checked and "Password Never Expires" check box is checked.

   e. Click Groups to make the user a member of the Administrator’s group.

The Group Memberships dialog box appears:



f. Double-click the Administrators icon from the window on the right to move it to the Member of list box.

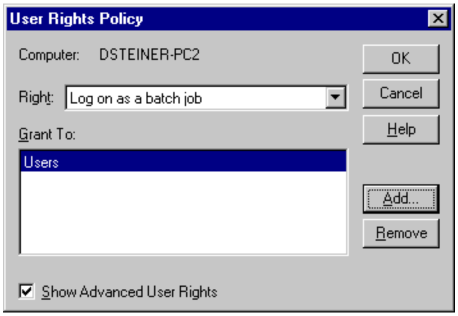
g. Click OK to close the Group Memberships dialog box and return to the New User dialog box.

4. Click OK in the New User dialog box.

   The user is added and the User Manager window re-appears.

5. Select the newly-created user, then choose Policies > User Rights.

   The User Rights Policy dialog box appears:



Computer: DSTEINER-PC2

Right: Log on as a batch job

Grant To: Users

☑ Show Advanced User Rights

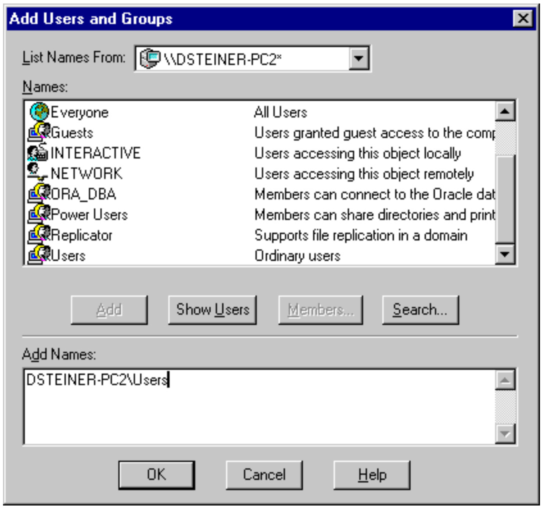
6. Click the Show Advanced User Rights check box, then select Log on as a batch job from the Right drop-down list box.

7. Select the user name from the Grant To list box.

   If the user is not listed, create it. To do this:

a. Click Add.

The Add Users and Groups dialog box appears:



Add Users and Groups

List Names From: \\DSTEINER-PC2*

Names:

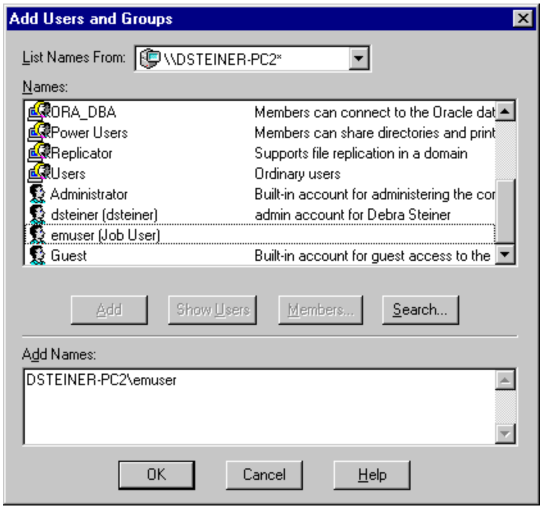
Everyone	All Users
Guests	Users granted guest access to the comp
INTERACTIVE	Users accessing this object locally
NETWORK	Users accessing this object remotely
ORA_DBA	Members can connect to the Oracle dat
Power Users	Members can share directories and print
Replicator	Supports file replication in a domain
Users	Ordinary users

Add Names: DSTEINER-PC2\Users

b. Select the name of your local host machine from the List Names From drop-down list box.

c. Click the Show Users button.

d. Select the user from the Names list box, then click Add:



List Names From: \\DSTEINER-PC2*

Names:

ORA_DBA	Members can connect to the Oracle dat
Power Users	Members can share directories and print
Replicator	Supports file replication in a domain
Users	Ordinary users
Administrator	Built-in account for administering the cor
dsteiner (dsteiner)	admin account for Debra Steiner
emuser (Job User)	
Guest	Built-in account for guest access to the

Add Names:

DSTEINER-PC2\emuser

The new user name appears in the Add Names list box.

e. Click OK.

The user appears in the Grant To list box in the User Rights Policy dialog box.

8. Click OK in the User Rights Policy dialog box in the User. Control returns to the User Manager window.

9. Choose User > Exit.

Grant SYSDBA or SYSOPER Privileges to a User

Identify a current user, such as SYSTEM, or create a new user that is to be used to 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:

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

Set User Credentials in the Console

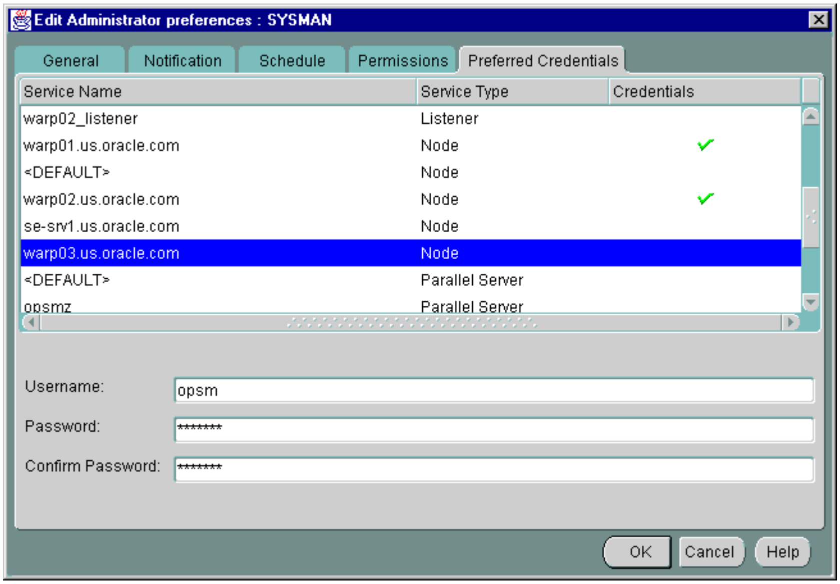
You must set user credentials for the database and each node.

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:



Service Name	Service Type	Credentials
warp02_listener	Listener	
warp01.us.oracle.com	Node	✓
<DEFAULT>	Node	
warp02.us.oracle.com	Node	✓
se-srv1.us.oracle.com	Node	
warp03.us.oracle.com	Node	
<DEFAULT>	Parallel Server	
opsmz	Parallel Server	

Username: opsm

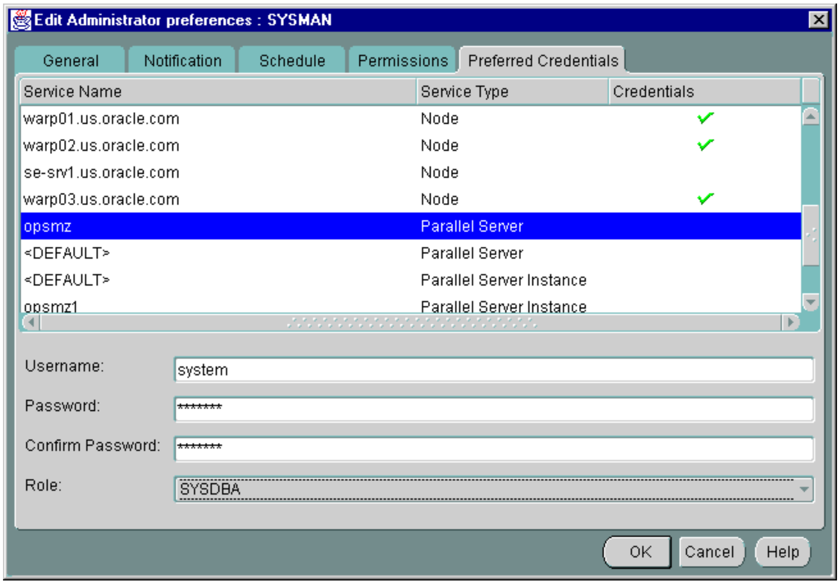
Password: *******

Confirm Password: *******

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 Oracle Parallel Server database (identified by the Parallel Server service type):



Service Name	Service Type	Credentials
warp01.us.oracle.com	Node	✓
warp02.us.oracle.com	Node	✓
se-srv1.us.oracle.com	Node	
warp03.us.oracle.com	Node	✓
opsmz	Parallel Server	
<DEFAULT>	Parallel Server	
<DEFAULT>	Parallel Server Instance	
opsmz1	Parallel Server Instance	

Username: system

Password: *******

Confirm Password: *******

Role: SYSDBA

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

Important: The SYSDBA or SYSOPER privilege is required for the Oracle Parallel Server database and instance startup and shutdown.

7. If you plan to use the Oracle Performance Manager application, click on an instance (identified by the Parallel Server Instance service type), enter a user name and password that can connect to the instance, select NORMAL role from the drop-down menu.

8. Click OK.

See Also:

- "Overview" on page 7-2 for more information about using the Console for Oracle Parallel Server
- *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 and the Oracle Intelligent Agent are *not* required by the Oracle Performance Manager, as its functions are performed using a database connection to the Oracle Parallel Server.

To configure Oracle Performance Manager, perform these tasks:

Task 1: Start Oracle Performance Manager Components

Task 2: Accessing Oracle Parallel Server Charts

Task 1: Start Oracle Performance Manager Components

To use the Oracle Performance Manager, start the following components:

- Oracle Data Gatherer
- Oracle Performance Manager

Oracle Data Gatherer

Start the Oracle Data Gatherer on at least one of the Oracle Parallel Server nodes.

At an operating system prompt, enter:

```
vppcntl -start
```

On Windows NT, you can also use the Control Panel:

1. Double-click the Services icon in the Control Panel window.
2. Select the Oracle*HOME_NAME*DataGatherer service.
3. Click Start to start the service.

Oracle Performance Manager

1. Start Oracle Performance Manager in one of two ways:

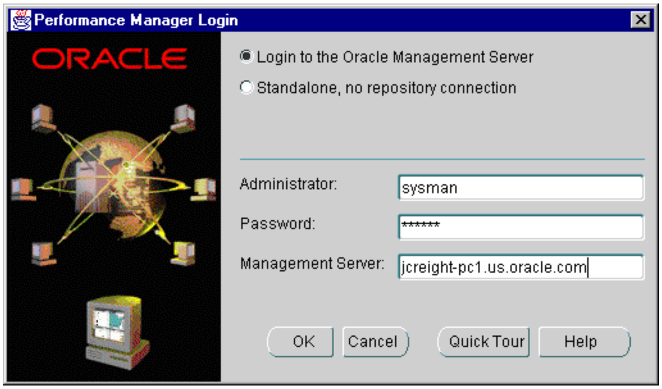
   - Standalone

     At an operating system prompt, enter:

     ```
     oemapp pm
     ```

     On Windows NT, you can also choose Start > Programs > Oracle - *HOME_NAME* > Oracle Diagnostics Pack > Performance Manager.

     The Performance Manager Login dialog box appears:

     Performance Manager Login

     ● Login to the Oracle Management Server

     ○ Standalone, no repository connection

     Administrator: sysman

     Password: ******

     Management Server: jcreight-pc1.us.oracle.com

     Select whether to log on to the Management Server or to log on in standalone mode. If "Login to Oracle Management Server" is chosen, additionally enter the Oracle Enterprise Manager administrator user name, password and location of the Management Server.

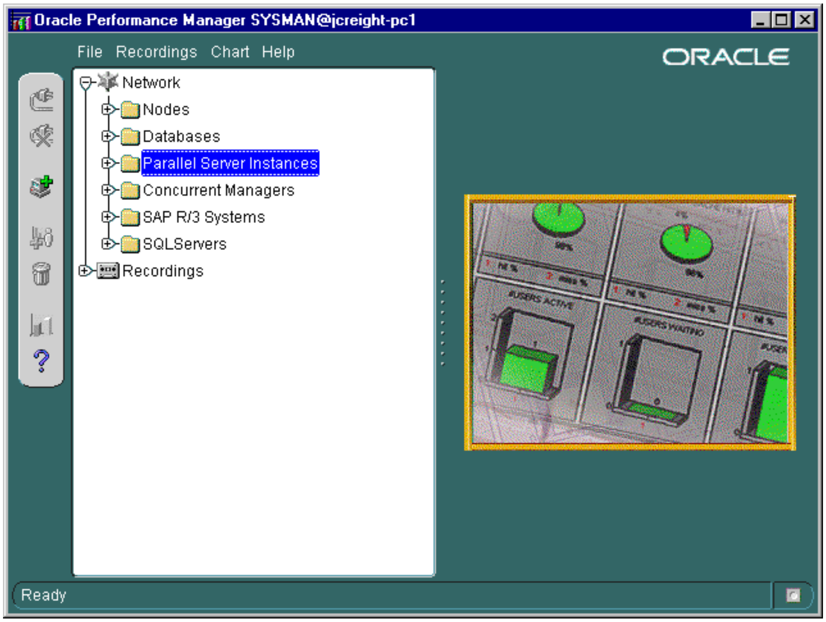
   - Started from Console

     Choose Oracle Performance Manager in the Diagnostics Pack toolbar or choose Tools > Diagnostics Pack and click Oracle Performance Manager.

     When you start Oracle Performance Manager from the Console, the Oracle Enterprise Manager repository log on credentials are automatically passed to Oracle Performance Manager. This provides you with access to all repository-based information. Also, if a service is selected in the Console Navigator window when Oracle Performance Manager is started, Oracle Performance Manager connects to the database.

2. Click OK to start Oracle Performance Manager.

   The Oracle Performance Manager starts:


Oracle Performance Manager SYSMAN@jcreight-pc1
File Recordings Chart Help
ORACLE
Network
Nodes
Databases
Parallel Server Instances
Concurrent Managers
SAP R/3 Systems
SQLServers
Recordings
Ready

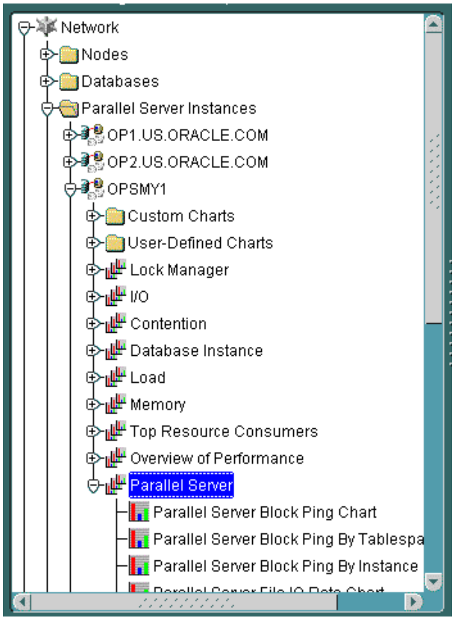
Task 2: Accessing Oracle Parallel Server Charts

You can access Oracle Parallel Server charts from the Parallel Server Instance folder or from the Databases folder.

Parallel Server Instances Folder

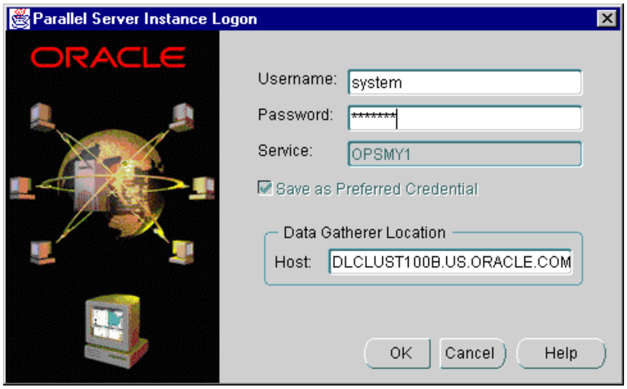
If the log in occurred from the Console or the "Login to the Oracle Management Server" option was selected in the Performance Manager Login dialog box, you can expand the Parallel Server Instances folder to display the instances from which the list of available charts may be obtained:.

Once you expand the Parallel Server Instances folder, expand User Defined Charts > Parallel Server.



Network
Nodes
Databases
Parallel Server Instances
OP1.US.ORACLE.COM
OP2.US.ORACLE.COM
OPSMY1
Custom Charts
User-Defined Charts
Lock Manager
I/O
Contention
Database Instance
Load
Memory
Top Resource Consumers
Overview of Performance
Parallel Server
Parallel Server Block Ping Chart
Parallel Server Block Ping By Tablespa
Parallel Server Block Ping By Instance
Parallel Server File IO Rate Chart

If you did not specify credentials for the instances, as described in "Task 4: Specify Preferred Credentials for Nodes and Oracle Parallel Server Database" on page 6-19, the Parallel Server Instances Logon dialog appears when you attempt to expand an instance:



Parallel Server Instance Logon

Username: system

Password: *******

Service: OPSMY1

☑ Save as Preferred Credential

Data Gatherer Location

Host: DLCLUST100B.US.ORACLE.COM

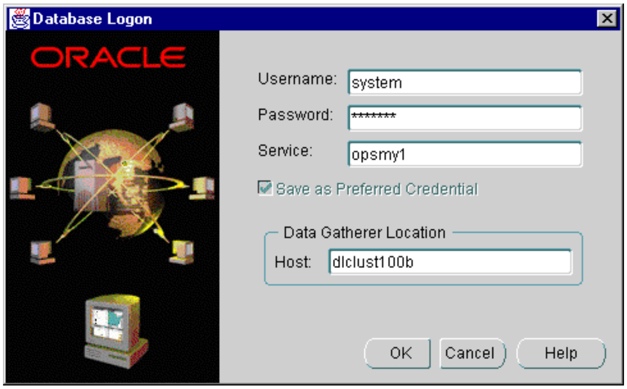
Enter a database user and password. The Oracle Database Gatherer location is automatically specified.

Databases Folder

If the log in 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 Oracle Parallel Server 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 credentials:



Database Logon

Username: system

Password: *******

Service: opsmy1

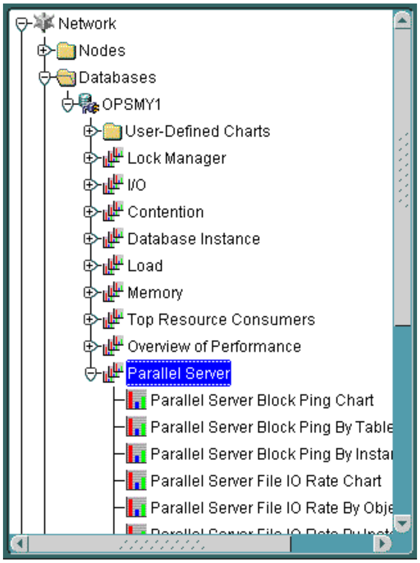
☑ Save as Preferred Credential

Data Gatherer Location

Host: dlclust100b

3. Enter the required information, including a database user name and password, a net service name in the Service field, and the location of the Oracle Data Gatherer. The Oracle Data Gatherer should be installed and running on one of the nodes in the cluster.

4. Expand Databases > User Defined Charts > Parallel Server.


Network
Nodes
Databases
OPSMY1
User-Defined Charts
Lock Manager
I/O
Contention
Database Instance
Load
Memory
Top Resource Consumers
Overview of Performance
Parallel Server
Parallel Server Block Ping Chart
Parallel Server Block Ping By Table
Parallel Server Block Ping By Insta
Parallel Server File IO Rate Chart
Parallel Server File IO Rate By Obje

See Also:

- Chapter 8, "Monitoring Performance with Oracle Performance Manager" for more information about viewing Oracle Parallel Server reports
- *Getting Started with the Oracle Standard Management Pack* for general information about the Oracle Performance Manager application

Additional Notes for DBA Studio

If you plan to run DBA Studio applications 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.

> **See Also:** "Net Service Names (tnsnames.ora file)" on page 3-16 for an example `tnsnames.ora` file

When running the DBA Studio application in stand-alone mode, the Startup and Shutdown menu items start and stop an individual instance, not the Oracle Parallel Server database.

7

Administering Oracle Parallel Server with Oracle Parallel Server Management

This chapter describes how multiple instances are managed through initialization files and Oracle Parallel Server Management.

Specific topics covered in this chapter are:

Overview

Oracle Parallel Server Management allows you to perform a variety of management tasks on your Oracle Parallel Servers, distributed systems, and databases. Oracle Enterprise Manager Console provides a central point of control for the Oracle environment through an intuitive graphical user interface (GUI) that provides drag-and-drop system management.

The Console enables you to manage a heterogeneous environment as easily as a homogeneous one. You can schedule jobs on multiple nodes simultaneously or monitor groups of services together.

A database or system administrator can thus control all nodes in an Oracle Parallel Server as a single entity. For example, you can configure Oracle Enterprise Manager to execute a job across all nodes of an Oracle Parallel Server.

This section is meant to only describe Oracle Enterprise Manager administration for Oracle Parallel Server. Use this section as a supplement to general information contained in the *Oracle Enterprise Manager Administrator's Guide*.

See Also: Chapter 5, "Configuring High-Availability Features" for configuration information

Starting the Console

To use the Console, start the following components:

- Oracle Intelligent Agents on each of the nodes
- Management Server
- Console

See Also: "Task 2: Start Oracle Enterprise Manager Components" on page 6-10 for instructions

Displaying Oracle Parallel Server Objects in the Navigator Window

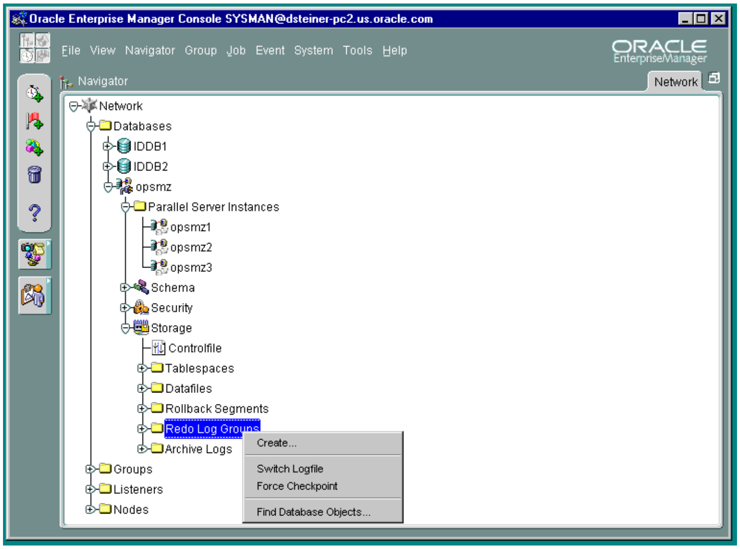
From the Navigator window, you can view and manage both single- and multiple-instance databases. The information available for Oracle Parallel Servers is the same as for single-instance databases.

The Navigator displays all the network objects and their relationships to other objects including a direct view of objects such as user-defined groups, nodes, listeners, servers, databases, and database objects.

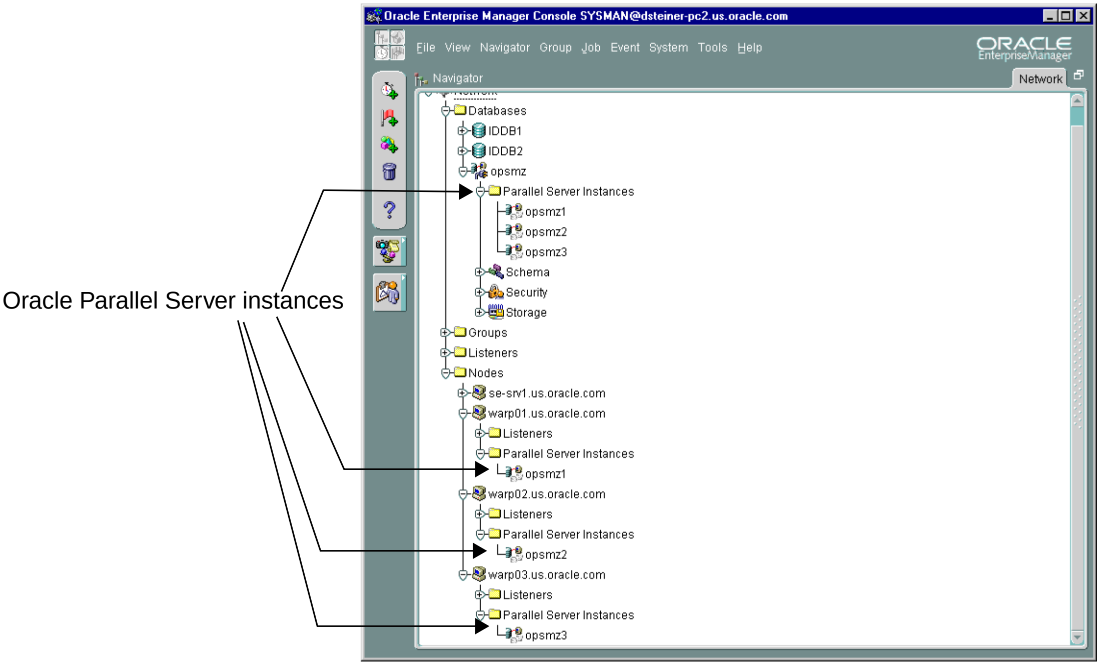
From the Navigator window, Oracle Parallel Servers are located in the Databases folder which contains entries for single-instance and Oracle Parallel Server databases. Each Oracle Parallel Server folder contains the instances and sub-folders for schema, security and file storage.

Note: If you did not set preferred credentials for the database, as described in "Task 4: Specify Preferred Credentials for Nodes and Oracle Parallel Server Database" on page 6-19, the Database Connect Information dialog prompts you to enter database connect information.

An Oracle Parallel Server database's subfolders behave just as they do for single-instance databases. By right-clicking the mouse on these objects, property sheets can be accessed to inspect and modify properties of these objects just as for single-instance databases. For example, right-clicking on the Redo Log Groups folder and choosing Create can add a new redo log group.

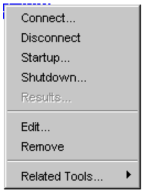


The Oracle Parallel Server database and each discovered Oracle Parallel Server node contains a Parallel Server Instances folder that you can expand to display the instances belonging to the database or node.


Oracle Enterprise Manager Console SYSMAN@dsteiner-pc2.us.oracle.com
File View Navigator Group Job Event System Tools Help
ORACLE EnterpriseManager
Navigator
Network
Databases
IDDB1
IDDB2
opsmz
Parallel Server Instances
opsmz1
opsmz2
opsmz3
Schema
Security
Storage
Groups
Listeners
Nodes
se-srv1.us.oracle.com
warp01.us.oracle.com
Listeners
Parallel Server Instances
opsmz1
warp02.us.oracle.com
Listeners
Parallel Server Instances
opsmz2
warp03.us.oracle.com
Listeners
Parallel Server Instances
opsmz3
Oracle Parallel Server instances

Using the Right-Mouse Menu

Right-clicking on a Oracle Parallel Server database displays the Oracle Parallel Server right-mouse menu:



This menu contains entries for the following functions:

Option	Description
Connect	Connects to the database. In the *Database Connect Information* dialog box, enter the database user name, password and privileges. If you want to perform a startup or shutdown, you must specify a user with SYSDBA privileges.
Disconnect	Terminates a connection to the database
Startup	Starts the database **See Also:** "Starting an Oracle Parallel Server Database"
Shutdown	Shuts down the database **See Also:** "Shutting Down an Oracle Parallel Server Database" on page 7-9
Results	Displays startup and shutdown results **See Also:** "Viewing the Parallel Server Operation Results" on page 7-12.
Edit	Allows inspection of the state of the Oracle Parallel Server, including which instances are active. **See Also:** "Viewing Oracle Parallel Server Status" on page 7-16
Remove	Deletes the database object and its related services. This should only be performed if it is no longer necessary to monitor or manage a database from Oracle Enterprise Manager.
Related Tools	Contains access to other tools which have been enabled for Oracle Parallel Server

Starting an Oracle Parallel Server Database

The Console allows you to start an Oracle Parallel Server database or individual instances.

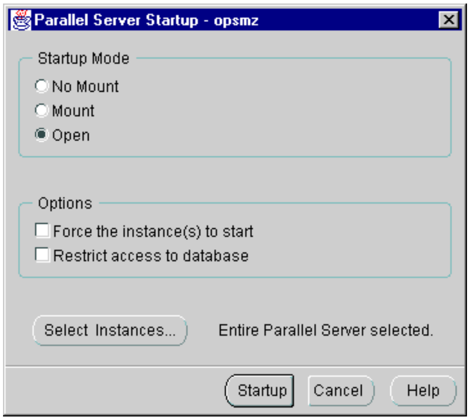
Once all Oracle Parallel Server instances are started, the Oracle Parallel Server database is considered to be up.

To start up an Oracle Parallel Server database:

1. In the Navigator window, expand Databases.
2. Right-click on an Oracle Parallel Server database.

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

   The Parallel Server Startup dialog box appears.



Parallel Server Startup - opsmz

Startup Mode

○ No Mount

○ Mount

● Open

Options

□ Force the instance(s) to start

□ Restrict access to database

Entire Parallel Server selected.

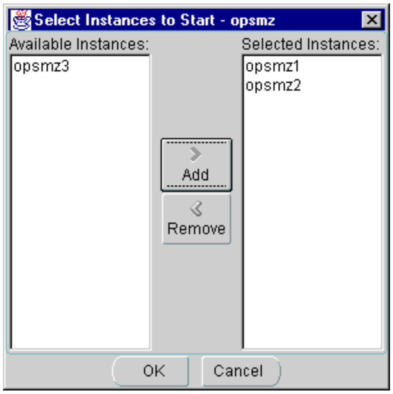
4. Select the startup type:

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

5. If you want to start up all instances, click Startup. If you want to start up only selected instances, follow these steps:

   a. Select Instances.

      The Select Instances to Start dialog box appears:



Select Instances to Start - opsmz

Available Instances: opsmz3

Selected Instances: opsmz1, opsmz2

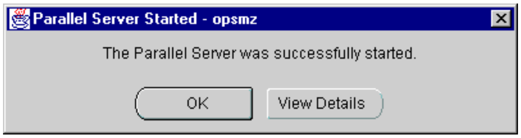
   b. Select the instances to start up in the Available list, then click Add.

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

d. Click Startup from the Parallel Server Startup dialog box.

The Parallel Server Startup Results dialog box displays the progress of the startup operation, as described in "Viewing the Parallel Server Operation Results" on page 7-12.

If the instances were started successfully, the Parallel Server Started message box appears with a successful message:


Parallel Server Started - opsmz

The Parallel Server was successfully started.

Click OK in the Parallel Server Started message to acknowledge the message, then click Close in the Parallel Server Startup Results.

If the startup fails, the Parallel Server Started message box appears with a failure message. Click View Details to view more information in the Parallel Server Startup Results dialog box about why the startup failed, then click Close.

Shutting Down an Oracle Parallel Server Database

The Console allows you to shut down an Oracle Parallel Server database or individual instances.

Once all Oracle Parallel Server instances are shut down, the Oracle Parallel Server is also considered to be shut down.

Note: Occasionally, an Oracle Parallel Server database may be completely down, but some of its services, such as the database listener, may remain running.

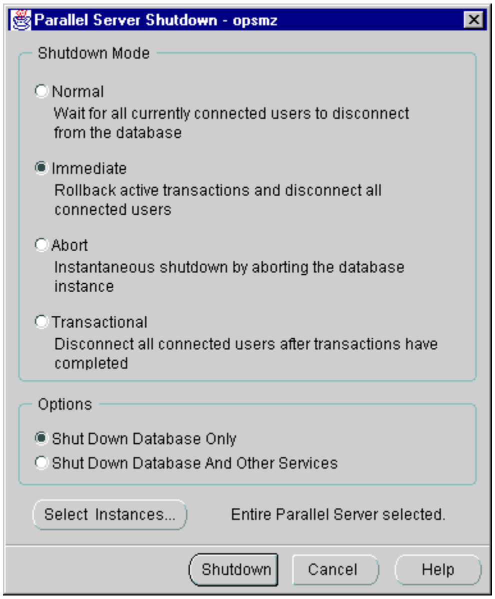
To shut down an Oracle Parallel Server database:

1. In the Navigator window, expand Databases.
2. Right-click on an Oracle Parallel Server database.

   A menu appears with options for the database.

3. Choose Shutdown from the menu.

   The Parallel Server Shutdown dialog box appears.



Parallel Server Shutdown - opsmz

Shutdown Mode

○ Normal
Wait for all currently connected users to disconnect from the database

● Immediate
Rollback active transactions and disconnect all connected users

○ Abort
Instantaneous shutdown by aborting the database instance

○ Transactional
Disconnect all connected users after transactions have completed

Options

● Shut Down Database Only
○ Shut Down Database And Other Services

Entire Parallel Server selected.

4. Select the shutdown type:

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

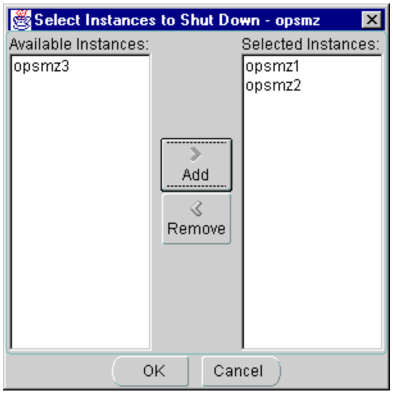
Option	Description
Transactional	Waits for transaction to complete before shutting down
Shutdown Database Only	(*default*) Shuts down the database only. The services required for an instance, such as the listener, remain up and available.
Shutdown Database And Other Services	Shuts down the database and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

5. To shut down all instances, click Shutdown.

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

   a. Select Instances.

      The Select Instances to Stop dialog box appears:



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

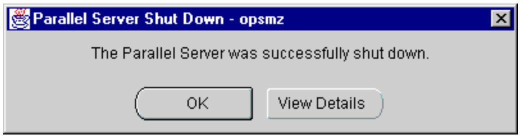
   c. Click OK to close the Select Instances to Stop dialog box.

   d. Click Shutdown from the Parallel Server Shutdown dialog box.

   The Parallel Server Shutdown Progress dialog box displays the progress of the shutdown operation.

See Also: "Viewing the Parallel Server Operation Results" on page 7-12

If the instances were shut down successfully, the Parallel Server Stopped message box appears with a successful message.:



Parallel Server Shut Down - opsmz

The Parallel Server was successfully shut down.

Click OK in the Parallel Server Stopped message to acknowledge the message, then click Close in the Parallel Server Shutdown Results.

If the shutdown fails, the Parallel Server Stopped message box appears with a failure message. Click View Details to view more information in the Parallel Server Shutdown Progress dialog box about why the shutdown failed, then click Close.

Viewing the Parallel Server Operation Results

The Parallel Server Startup / Shutdown Results dialog displays information about the progress of the instance startup or shutdown operation you selected:

The operation results are presented in two views:

- Status Details Tab
- Output Tab

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

1. In the Navigator window, expand Databases.
2. Right-click on an Oracle Parallel Server database.

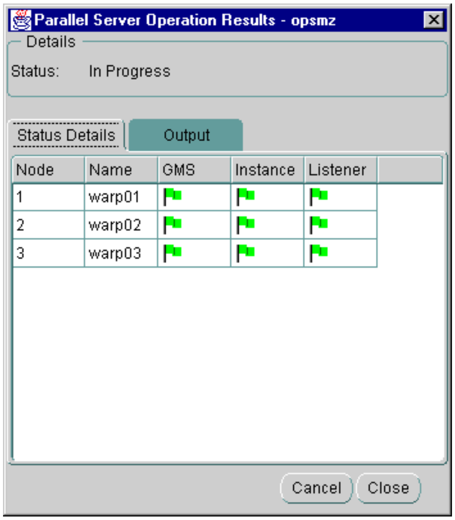
   A menu appears with options for the database.
3. Choose Results from the menu.

Status Details Tab

> **Note:** This tab is not available for Windows NT Oracle Parallel Servers, because OPSCTL on Windows NT does not generate status details.

While a startup or shutdown operation is running against an Oracle Parallel Server, the Status Details tab progress display is shown and updated dynamically as the operation progresses.

A successful startup operation for a three-node cluster looks like the following in the Status Details tab:

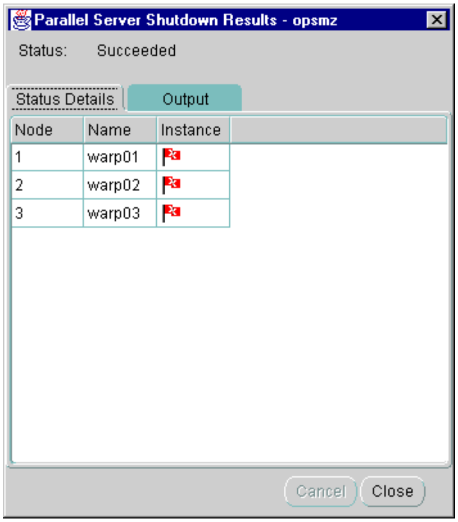


Status: In Progress

Node	Name	GMS	Instance	Listener
1	warp01	⚑	⚑	⚑
2	warp02	⚑	⚑	⚑
3	warp03	⚑	⚑	⚑

The services managed by Oracle Parallel Server Management vary by operating system and by version. The Oracle Parallel Server being managed in this example is release 8.0.5, so the GMS (Group Membership Service) is being started. In release 8.1, the GMS is built into the RDBMS kernel, so only the instance and listener columns display.

A successful shutdown operation for a three-node cluster looks like the following in the Status Details tab:



Parallel Server Shutdown Results - opsmz

Status: Succeeded

Node	Name	Instance
1	warp01	🚩
2	warp02	🚩
3	warp03	🚩

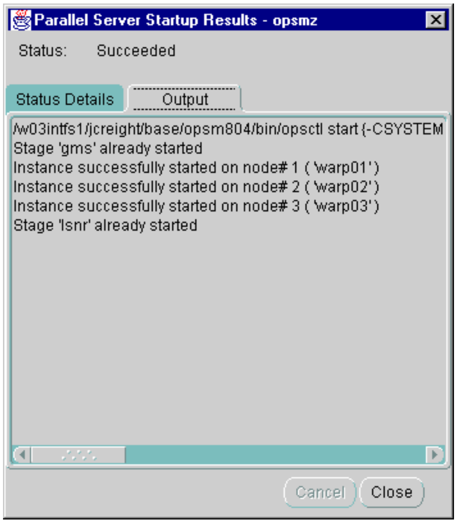
The following are the possible states that each component may experience:

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

Output Tab

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

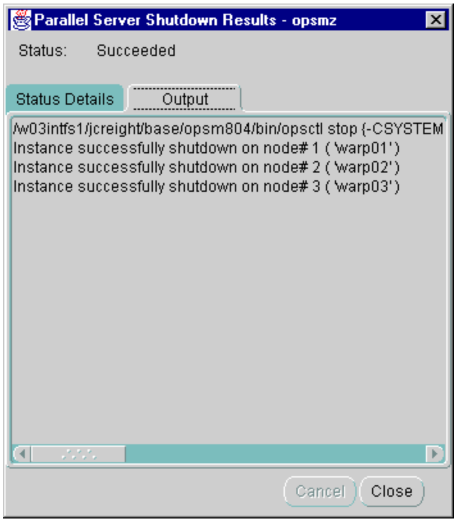
A successful startup for a three-node cluster looks like the following in the Output tab:



Status: Succeeded

```
/w03intfs1/jcreight/base/opsm804/bin/opsctl start {-CSYSTEM
Stage 'gms' already started
Instance successfully started on node# 1 ( 'warp01' )
Instance successfully started on node# 2 ( 'warp02' )
Instance successfully started on node# 3 ( 'warp03' )
Stage 'lsnr' already started
```

A successful shutdown operation for a three-node cluster looks like the following in the Output tab:



Status: Succeeded

```
/w03intfs1/jcreight/base/opsm804/bin/opsctl stop {-CSYSTEM
Instance successfully shutdown on node# 1 ( 'warp01' )
Instance successfully shutdown on node# 2 ( 'warp02' )
Instance successfully shutdown on node# 3 ( 'warp03' )
```

Viewing Oracle Parallel Server Status

The Edit Parallel Server dialog box displays status information about the Oracle Parallel Server database, such as instances available in the Oracle Parallel Server and status of Oracle Parallel Server components.

Note: Because this dialog box requires a connection to an Oracle Parallel Server, this dialog box will not appear if the Oracle Parallel Server is down.

To view status information about an Oracle Parallel Server database:

1. In the Navigator window, expand Databases > *database_name*.
2. Right-click on an Oracle Parallel Server database under the Databases folder in the Navigator window.

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

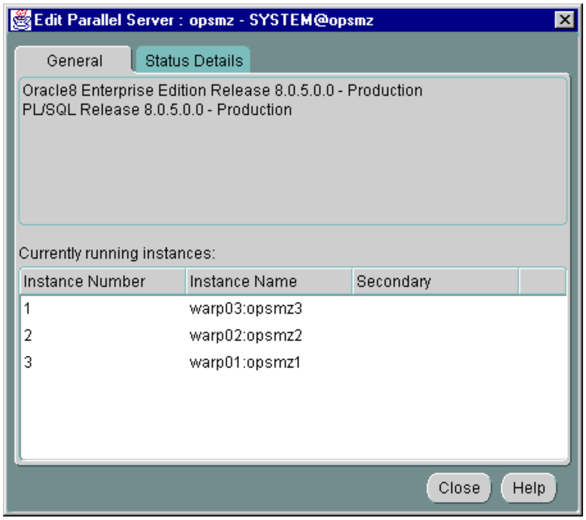
   The Edit Parallel Server dialog box appears.

The operation results are presented in two views:

- General Tab
- Status Details Tab

General Tab

The Status tab displays information about the currently running instances by querying V$ACTIVE_INSTANCES table. Oracle Enterprise Manager makes a connection to the Oracle Parallel Server; therefore, this tab will not appear if the Oracle Parallel Server is down.



Oracle8 Enterprise Edition Release 8.0.5.0.0 - Production
PL/SQL Release 8.0.5.0.0 - Production

Currently running instances:

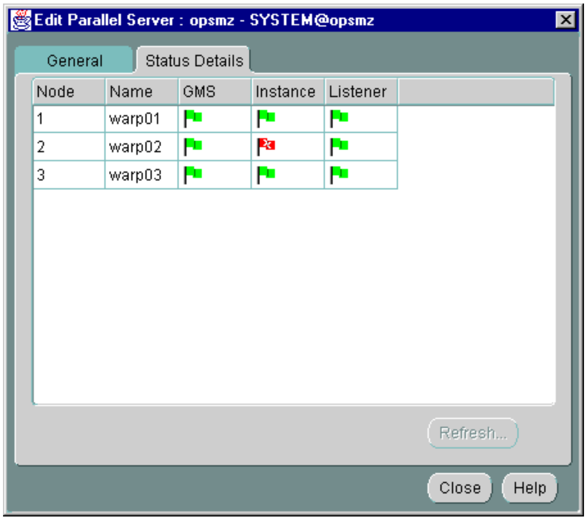
Instance Number	Instance Name	Secondary
1	warp03:opsmz3	
2	warp02:opsmz2	
3	warp01:opsmz1	

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

Status Details Tab

Note: This tab is not available for a Windows NT Oracle Parallel Server, because OPSCTL on Windows NT does not generate status details.

Displays an overall view of the state of the Oracle Parallel Server and related components. This tab displays the status of the various components, such as listeners and instances, for all nodes.



Node	Name	GMS	Instance	Listener
1	warp01	🟩	🟩	🟩
2	warp02	🟩	🟥	🟩
3	warp03	🟩	🟩	🟩

The following are the possible states that each component may experience:

State	Description
Up (green flag)	The component is running.
Down (red flag)	The component is not running.
In Progress (timer)	Oracle Enterprise Manager cannot determine the state of the component. This state occurs typically when the component startup or shutdown operation has not completed.

State	Description
Component does not exist on this node (blank background)	The component was not configured on the node. Not all components (listener, instance) are required to exist on every node.

Creating a Job for a Parallel Server or Parallel Server Instance

The job scheduling system provides a highly reliable and flexible mechanism for DBAs to schedule and automate repetitive jobs on both the Oracle Parallel Server database and Oracle Parallel Server instances.

The Console contains a full-featured scheduling tool that allows DBAs to develop a customized schedule. This provides DBAs with actual “lights out” management capability so the DBAs can focus on other tasks. A rich selection of jobs is provided for Oracle Parallel Servers.

You can create a job with an Oracle Parallel Server database or an Oracle Parallel Server instance as the destination. To create a new job, follow these steps:

1. Choose Job > Create Job.
2. Complete the tabs of the Create Job property sheet.
3. When you are satisfied with your job settings, click the Submit button to submit the job to Oracle Intelligent Agent. The job appears in the Active Jobs window.
4. Click the Save button to save the job. The job appears in the Job Library window. You can modify or submit a saved job at a later time.

Note: There is usually a slight delay between submitting the job and notification by Oracle Intelligent Agent.

Specifying Job Details

From the Create Job property sheet, you can specify the details of a new job. The Create Job property sheet contains these tabs:

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

The following tabs contain Oracle Parallel Server-specific options:

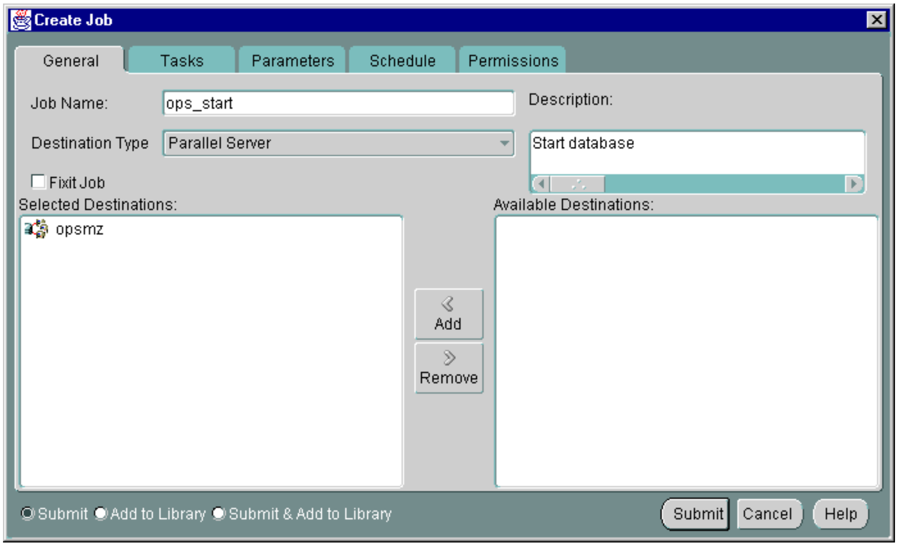
- General Tab
- Tasks Tab
- Parameters Tab

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

General Tab

From the General tab, specify the:

- Job Name
- Destination Type (either Parallel Server or Parallel Server Instance)
- Description
- Destinations



Job Name: ops_start

Destination Type: Parallel Server

Description: Start database

□ Fixit Job

Selected Destinations: opsmz

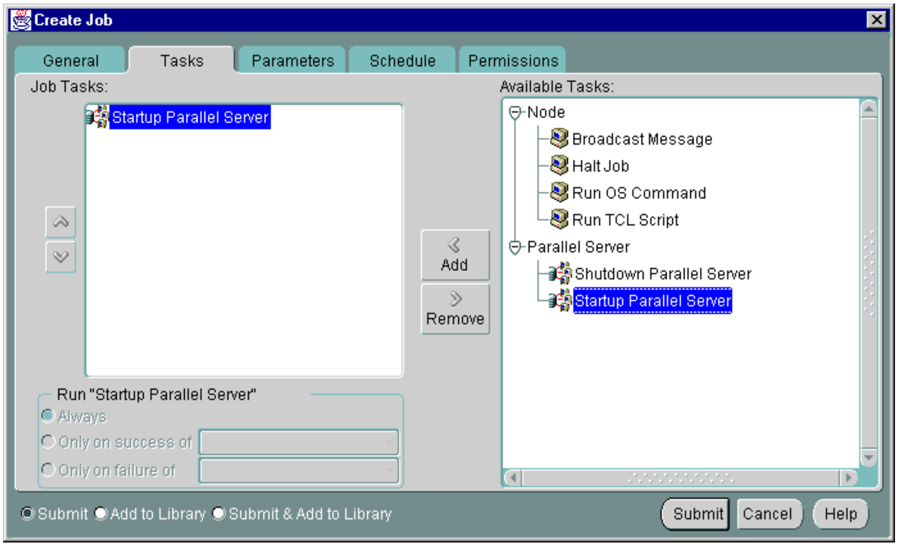
The General tab contains these options:

Parameter	Description
Job Name	Enter the name of the new job.
Description	Enter a description of the job.
Destination Type	Select a destination type from the drop-down list box. You can select from the following options: parallel server, parallel server instance, database, listener, or node.

Parameter	Description
Available Destinations	The destinations are determined by your selection of the Destination Type. The destinations include parallel servers, parallel server instances, databases, listeners, and nodes. Click the destinations of the job in the Available Destinations list, then click Add to move the destination to the Selected Destinations list. To remove a destination from a job, click the destination in the Selected Destinations list, then click Remove.
Fixit Job	Select this check box if you want to use this job as the fixit job for an event occurrence. The job cannot be scheduled.

Tasks Tab

From the Tasks tab, choose the task(s) that you want the job to perform. The list of tasks that appear is different depending on whether you select an Oracle Parallel Server or an Oracle Parallel Server instance as your Destination Type from the General tab.



Move the tasks between the Available Tasks and Selected Tasks lists with the Add and Remove buttons.

Tasks for Parallel Server Destinations

If your Destination Type is a Parallel Server, you can select from these tasks:

- Shutdown Parallel Server
- Startup Parallel Server

Tasks for Parallel Server Instance Destinations

If your Destination Type is a Parallel Server Instance, you can select from these tasks:

- Run DBA Script
- Run SQL*Plus Script
- Shutdown Database
- Startup Database

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

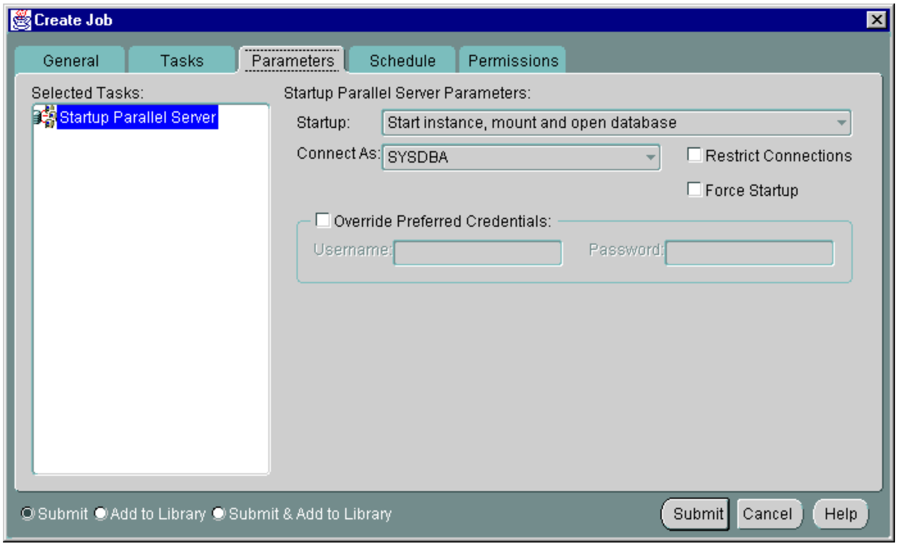
Parameters Tab

From the Parameters tab, you can specify parameter settings for the job tasks you selected on the Tasks tab. The parameters that display vary according to the job task. Parameters for Oracle Parallel Server startup and shutdown tasks are described below.

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

Parallel Server Startup Task

When you select the Startup Parallel Server task on the Tasks tab, the following display appears:



Selected Tasks: Startup Parallel Server

Startup Parallel Server Parameters:

Startup: Start instance, mount and open database

Connect As: SYSDBA

□ Restrict Connections

□ Force Startup

□ Override Preferred Credentials:

Username:

Password:

● Submit ○ Add to Library ○ Submit & Add to Library

Complete the parameters on the tab and click the Submit button to run the Oracle Parallel Server startup task.

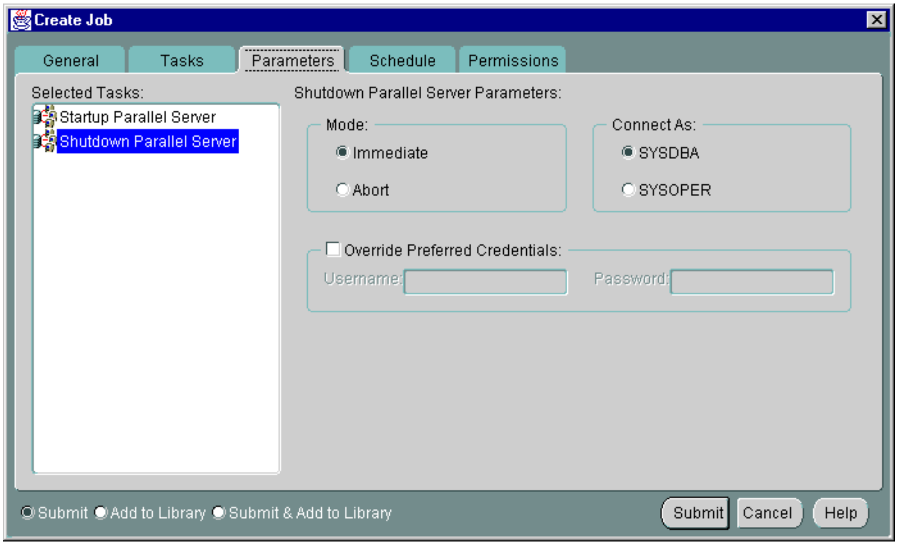
The Parameters tab for Startup contains these options:

Parameters	Description
Startup	Select the startup modes for the job from the drop-down list box.
Connect As	Specify the role. Only Normal is allowed for Oracle7. For Oracle8*i*, SYSOPER and SYSDBA roles allow you the maximum database administration privileges. You require SYSDBA or SYSOPER privileges to run job tasks such as shutdown or startup on the database. **See Also**: *Oracle8i Administrator's Guide* for more information about SYSOPER and SYSDBA roles If you attempt to connect as SYSDBA and do not have SYSDBA privileges, an error message states that an invalid user name or password was entered.

Parameters	Description
Override Preferred Credentials	You can use the preferred credentials that have been set up for the database, or you can enter another database user name and password.

Parameters for the Parallel Server Shutdown Task

When you select the Shutdown Parallel Server task on the Tasks tab, the following display appears.



Selected Tasks:
Startup Parallel Server
Shutdown Parallel Server

Shutdown Parallel Server Parameters:

Mode: ● Immediate ○ Abort

Connect As: ● SYSDBA ○ SYSOPER

□ Override Preferred Credentials:
Username:
Password:

○ Submit ○ Add to Library ○ Submit & Add to Library

Complete the parameter entries on the tab and click the Submit button to run the Oracle Parallel Server shutdown task.

The Parameters tab for Shutdown contains these options:

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

8

Monitoring Performance with Oracle Performance Manager

This chapter presents the Oracle Performance Manager performance and tuning charts specific to Oracle8*i* Parallel Server. You must have Oracle Performance Manager installed and configured to display the charts.

This chapter describes only Oracle Performance Manager features specific to Oracle Parallel Server. Use this chapters as a supplement to general information contained in the *Getting Started with the Oracle Standard Management Pack*.

See Also:

- Chapter 5, "Configuring High-Availability Features" for further information about installing Oracle Performance Manager
- *Oracle8i Parallel Server Administration, Deployment, and Performance* for further information about additional views not available with Oracle Enterprise Manager, and for more detail on monitoring and tuning Oracle Parallel Server
- *Oracle8i Designing and Tuning for Performance* for information about the statistics these charts display and how to interpret these statistics
- *Oracle8i Reference* for more information about the fields in these charts and the V$ views from which they are derived

This chapter covers the following topics:

- Oracle Performance Manager Overview
- Starting Oracle Performance Manager
- Displaying Charts

Oracle Performance Manager Overview

Oracle stores tuning and performance information for the Oracle Parallel Server in a set of dynamic performance tables known as the "V$ fixed views" Each active instance has its own set of fixed views. In Oracle Parallel Server, you can have Oracle Performance Manager query a global dynamic performance (GV$) view to retrieve the related V$ view information from all instances.

Oracle Performance Manager displays the retrieved information in a variety of tabular and graphic performance statistics for Oracle Parallel Server. The statistics represent the aggregate performance of all instances running on an Oracle Parallel Server. The statistics are displayed in individual charts and include information about data block pings, lock activity, file I/O, and session and user information. You can also use the Performance Manager to display an overview of all of these statistics on one chart.

Oracle Parallel Server performance monitoring is crucial for realizing the full potential of the system. There are several key performance metrics that you should constantly monitor to keep the Oracle Parallel Server in peak operating condition. The Oracle Performance Manager, available as an applet within Oracle Enterprise Manager, is an application designed to capture, compute, and present performance data that help database administrators focus on key performance metrics.

Oracle Parallel Server performance metrics are compiled into charts that are viewable with Oracle Performance Manager:

Chart	Description
Parallel Server Block Ping Chart	Displays the total block pings on the Oracle Parallel Server
Parallel Server Data Block Ping by Tablespace Chart	Displays the block pings per tablespace in the Oracle Parallel Server
Parallel Server Data Block Ping by Instance Chart	Displays the block pings per instance in the Oracle Parallel Server
Parallel Server File I/O Rate Chart	Displays the rate of physical reads and writes for all files in the Oracle Parallel Server database. You can drill down to obtain the same information either at the instance level or at the file level
Parallel Server File I/O Rate by Object Chart	Displays the rate of reads and writes per data file in the Oracle Parallel Server database

Chart	Description
Parallel Server File I/O Rate by Instance Chart	Displays the rate of reads and writes per instance in the Oracle Parallel Server database
Parallel Server Lock Activity Chart	Displays the statistics on the lock activity rate for all the different lock types across all Oracle Parallel Servers. You can drill down to obtain lock activity information for a particular lock type at the instance level
Parallel Server Sessions Chart	Displays the sessions attached to the Oracle Parallel Server and related information, such as instance name, session ID, session serial number, process ID, status, and user name
Parallel Server Users Logged On Chart	Displays the total number of user sessions logged on to the Oracle Parallel Server, regardless of whether activity is generated. This information is also available for each instance.
Parallel Server Users Logged On by Instance Chart	Displays the number of users logged in to the Oracle Parallel Server by instance
Parallel Server Active Users Chart	Displays the total number of active users on the Oracle Parallel Server
Parallel Server Active Users by Instance Chart	Displays the number of active user sessions
OPS Overview Chart	Displays a group of charts that display key performance statistics for the selected Oracle Parallel Server

Starting Oracle Performance Manager

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

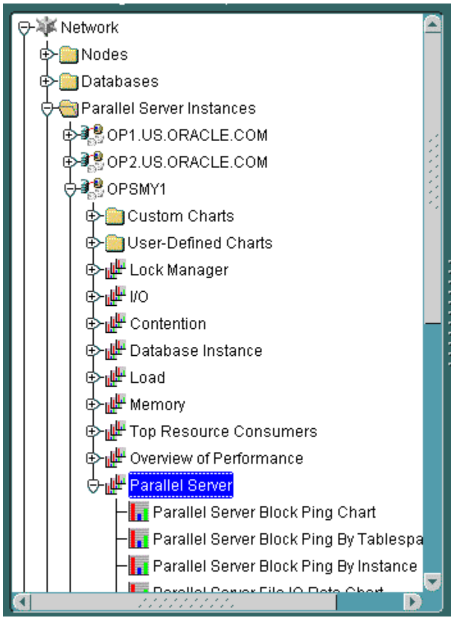
- Oracle Data Gatherer on one of the nodes
- Oracle Performance Manager

See "Task 1: Start Oracle Performance Manager Components" on page 6-27 for instructions.

Displaying Charts

To display charts:

1. Follow the procedures in "Task 1: Start Oracle Performance Manager Components" on page 6-27 and "Task 2: Accessing Oracle Parallel Server Charts" on page 6-30.
2. In the navigator, expand Databases or Parallel Server Instance > Parallel Server to see the list of available charts.



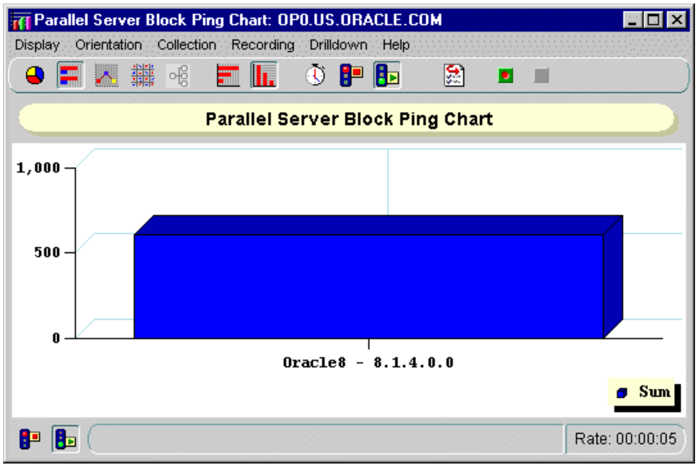
3. In the Parallel Server object folder, select a chart, then click Show Chart.

   The chart displays in a separate window.

Parallel Server Block Ping Chart

The Parallel Server Block Ping chart displays statistics from the GV$PING view on the total number of block pings:

Figure 8–1 Parallel Server Block Ping Chart



To display block pings at the tablespace or instance level:

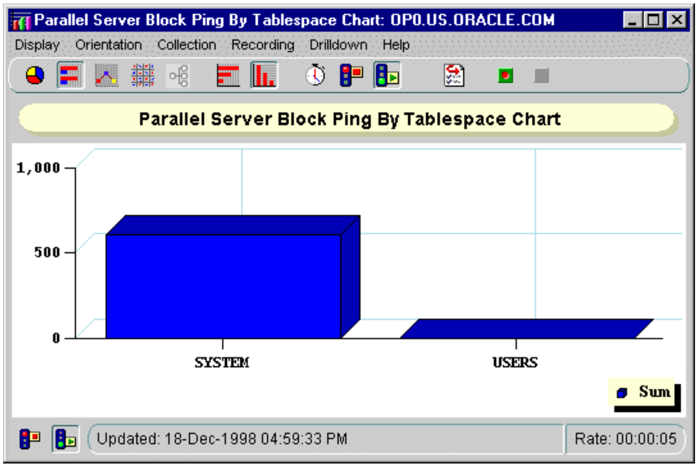
1. Click anywhere on the chart.
2. Right-click and select the chart from the Drilldown menu.

Chart	Description
Parallel Server Data Block Ping by Instance Chart	Displays pinging on individual instances
Parallel Server Data Block Ping by Tablespace Chart	Displays the pinging on the tablespaces

Parallel Server Data Block Ping by Tablespace Chart

The Parallel Server Block Ping by Tablespace chart displays statistics from the GV$PING view on the number of block pings by tablespace:

Figure 8–2 Parallel Server Block Ping by Tablespace Chart


Parallel Server Block Ping By Tablespace Chart: OP0.US.ORACLE.COM
Display
Orientation
Collection
Recording
Drilldown
Help
Parallel Server Block Ping By Tablespace Chart
1,000
500
0
SYSTEM
USERS
Sum

Updated: 18-Dec-1998 04:59:33 PM

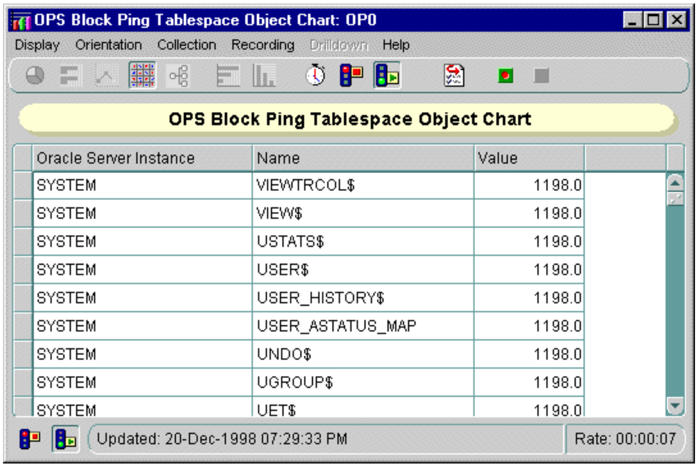
Rate: 00:00:05

To display block pings at the object or instance level:

1. Click on an individual tablespace.
2. Right-click and select the chart from the Drilldown menu:

Option	Description
Parallel Server Block Ping Tablespace Object Chart	Displays the objects for the selected tablespace. Objects represent tables, indexes, free extents, clusters of tables, and so on. Listed for each object is the number of pings in the tablespace in which that object resides.
Parallel Server Block Ping Tablespace Instance Chart	Displays the instances within the selected tablespace. Listed for each instance is the number of pings in the tablespace in which that instance resides.

Figure 8–3 OPS Block Ping Tablespace Object Chart

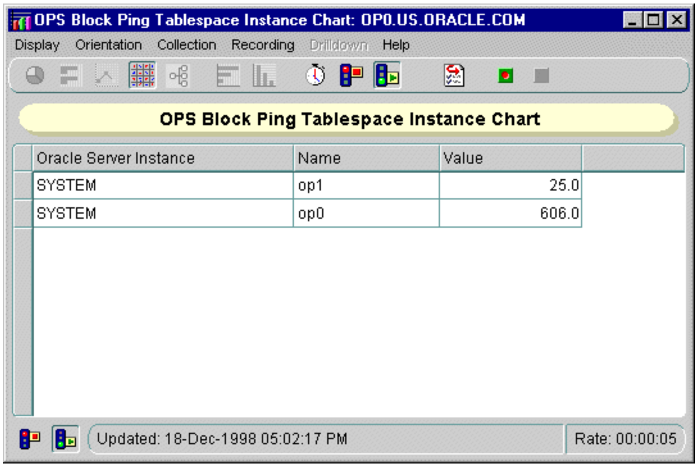

OPS Block Ping Tablespace Object Chart: OP0
Display Orientation Collection Recording Drilldown Help

OPS Block Ping Tablespace Object Chart

Oracle Server Instance	Name	Value
SYSTEM	VIEWTRCOL$	1198.0
SYSTEM	VIEW$	1198.0
SYSTEM	USTATS$	1198.0
SYSTEM	USER$	1198.0
SYSTEM	USER_HISTORY$	1198.0
SYSTEM	USER_ASTATUS_MAP	1198.0
SYSTEM	UNDO$	1198.0
SYSTEM	UGROUP$	1198.0
SYSTEM	UET$	1198.0

Updated: 20-Dec-1998 07:29:33 PM Rate: 00:00:07

Figure 8–4 OPS Block Ping Tablespace Instance Chart


OPS Block Ping Tablespace Instance Chart: OP0.US.ORACLE.COM
Display Orientation Collection Recording Drilldown Help

OPS Block Ping Tablespace Instance Chart

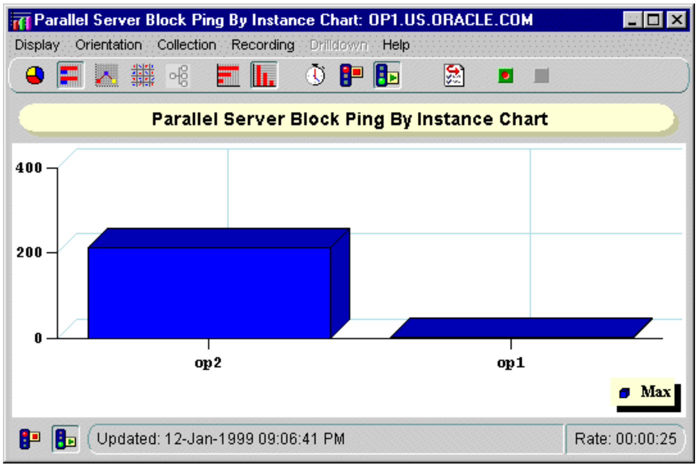
Oracle Server Instance	Name	Value
SYSTEM	op1	25.0
SYSTEM	op0	606.0

Updated: 18-Dec-1998 05:02:17 PM Rate: 00:00:05

Parallel Server Data Block Ping by Instance Chart

The Parallel Server Block Ping by Instance chart displays statistics from the GV$PING view on the number of block pings per instance:

Figure 8–5 Parallel Server Block Ping by Instance Chart


Parallel Server Block Ping By Instance Chart: OP1.US.ORACLE.COM
Display
Orientation
Collection
Recording
Drilldown
Help
Parallel Server Block Ping By Instance Chart
400
200
0
op2
op1
Max

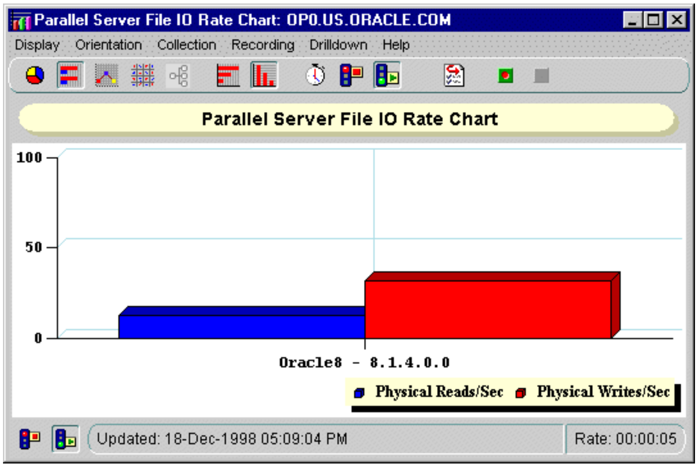
Updated: 12-Jan-1999 09:06:41 PM

Rate: 00:00:25

Parallel Server File I/O Rate Chart

The Parallel Server File I/O Rate chart displays physical files reads and writes for all files from the GV$FILESTAT view.

Figure 8–6 Parallel Server File IO Rate Chart



To display the file I/O rate at the instance level or at the file level:

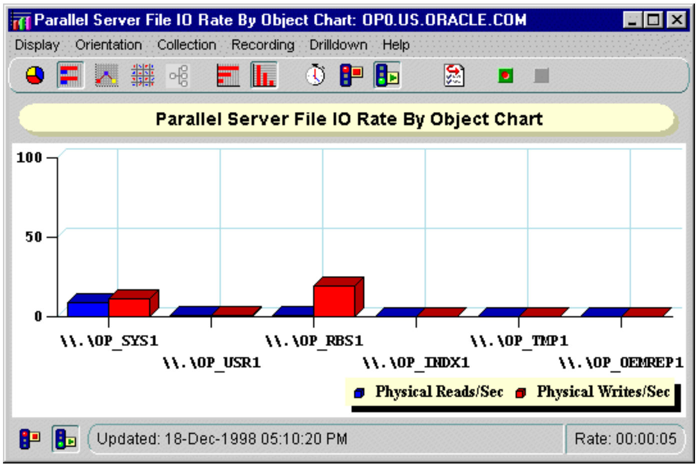
1. Click on a column's object.
2. Right-click and select the chart from the Drilldown menu.

Chart	Description
Parallel Server File I/O Rate by Instance Chart	Displays the file I/O rate on individual instances
Parallel Server File I/O Rate by Object Chart	Displays the file I/O rate on the tablespaces

Parallel Server File I/O Rate by Object Chart

The Parallel Server File I/O Rate by Object chart displays physical files reads and writes for individual data files from the GV$FILESTAT view.

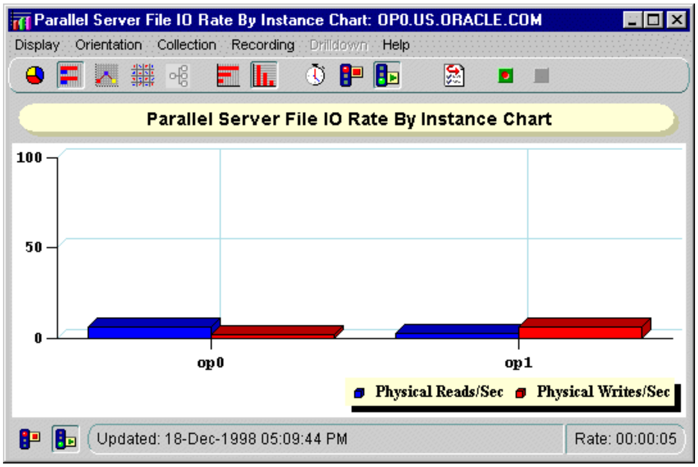
Figure 8–7 Parallel Server Block Ping by Object Chart



Parallel Server File I/O Rate by Instance Chart

The Parallel Server File I/O Rate by Instance chart displays physical files reads and writes for individual instances from the GV$FILESTAT view.

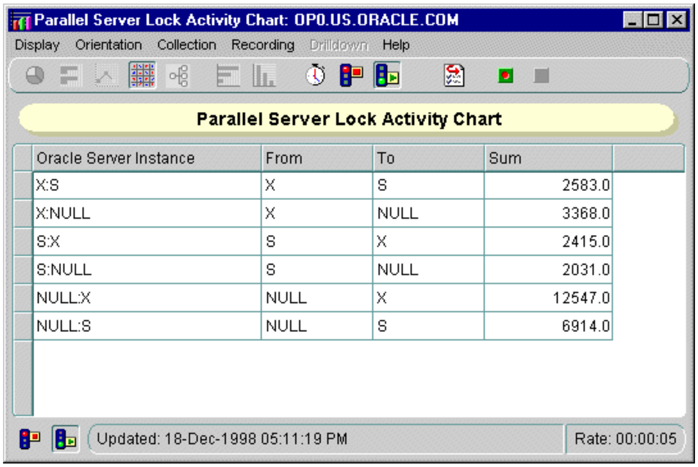
Figure 8–8 Parallel Server File IO Rate by Instance Chart


Parallel Server File IO Rate By Instance Chart: OP0.US.ORACLE.COM
Display
Orientation
Collection
Recording
Drilldown
Help
Parallel Server File IO Rate By Instance Chart
100
50
0
op0
op1
Physical Reads/Sec
Physical Writes/Sec
Updated: 18-Dec-1998 05:09:44 PM
Rate: 00:00:05

Parallel Server Lock Activity Chart

The Parallel Server Lock Activity chart shows the number of lock converts of various types (exclusive to null) from the GV$LOCK_ACTIVITY view:

Figure 8–9 Parallel Server Lock Activity Chart


Parallel Server Lock Activity Chart: OP0.US.ORACLE.COM
Display
Orientation
Collection
Recording
Drilldown
Help
Parallel Server Lock Activity Chart

Oracle Server Instance	From	To	Sum
X:S	X	S	2583.0
X:NULL	X	NULL	3368.0
S:X	S	X	2415.0
S:NULL	S	NULL	2031.0
NULL:X	NULL	X	12547.0
NULL:S	NULL	S	6914.0

Updated: 18-Dec-1998 05:11:19 PM

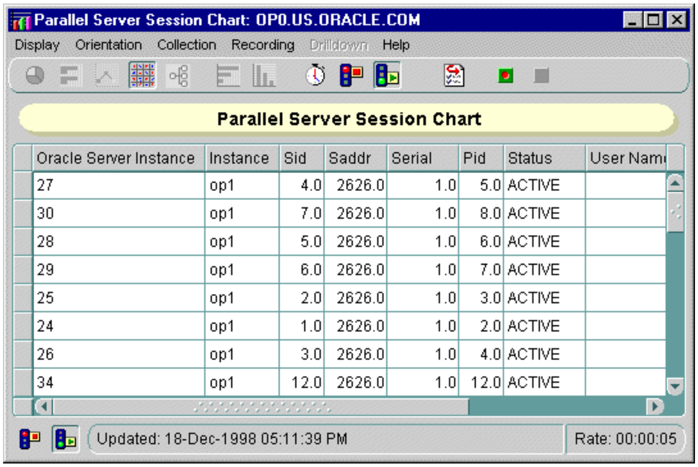
Rate: 00:00:05

Column	Description
From	PCM lock initial state: NULL; S; X; SSX
To	PCM lock final state: NULL; S; X; SSX
Sum	Displays the number of times the lock operation executed

Parallel Server Sessions Chart

The Parallel Server Sessions chart displays a table of Session IDs. This chart displays statistics from the GV$PROCESS and GV$SESSION and GV$PROCESS views.

Figure 8–10 Parallel Server Sessions Chart



Parallel Server Session Chart

Oracle Server Instance	Instance	Sid	Saddr	Serial	Pid	Status	User Nam
27	op1	4.0	2626.0	1.0	5.0	ACTIVE	
30	op1	7.0	2626.0	1.0	8.0	ACTIVE	
28	op1	5.0	2626.0	1.0	6.0	ACTIVE	
29	op1	6.0	2626.0	1.0	7.0	ACTIVE	
25	op1	2.0	2626.0	1.0	3.0	ACTIVE	
24	op1	1.0	2626.0	1.0	2.0	ACTIVE	
26	op1	3.0	2626.0	1.0	4.0	ACTIVE	
34	op1	12.0	2626.0	1.0	12.0	ACTIVE	

Updated: 18-Dec-1998 05:11:39 PM

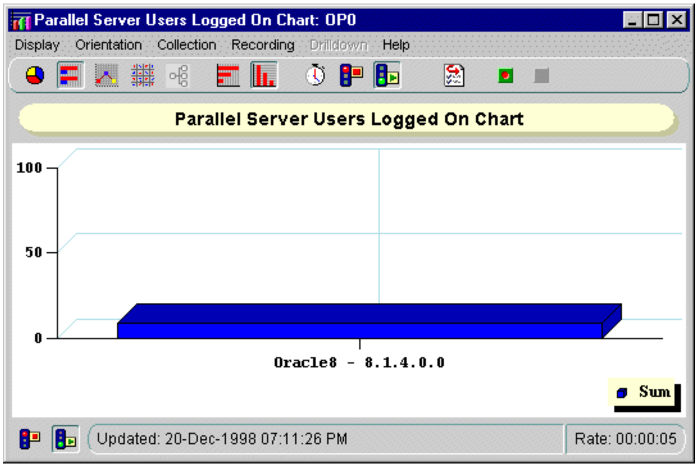
Rate: 00:00:05

Column	Description
SID	Displays the session identifier
Saddr	Displays the session address
Serial	Displays the session serial number. Used to uniquely identify a session's objects. Guarantees that session-level commands are applied to the correct session objects if the session ends and another session begins with the same session ID.
PID	Displays the operating system client process ID
Status	Displays the status of the session: ACTIVE (currently executing SQL), INACTIVE, KILLED (marked to be killed), CACHED (temporarily cached for use by Oracle*XA), SNIPED (session inactive, waiting on the client)
User Name	Displays the Oracle user name

Parallel Server Users Logged On Chart

The Parallel Server Users Logged On chart displays the total number of user sessions currently logged on to the Oracle Parallel Server, whether or not activity is being generated. This chart displays statistics from the GV$LICENSE view:

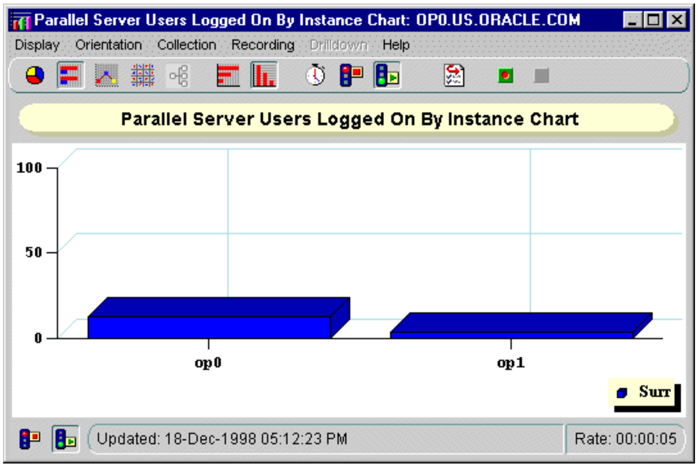
Figure 8–11 Parallel Server Users Logged On Chart


Parallel Server Users Logged On Chart: OPO
Display
Orientation
Collection
Recording
Drilldown
Help
Parallel Server Users Logged On Chart
100
50
0
Oracle8 - 8.1.4.0.0
Sum
Updated: 20-Dec-1998 07:11:26 PM
Rate: 00:00:05

Parallel Server Users Logged On by Instance Chart

The Parallel Server Users Logged On by Instance chart displays the number of users logged on to each instance in the Oracle Parallel Server. This chart displays statistics from the GV$LICENSE view.

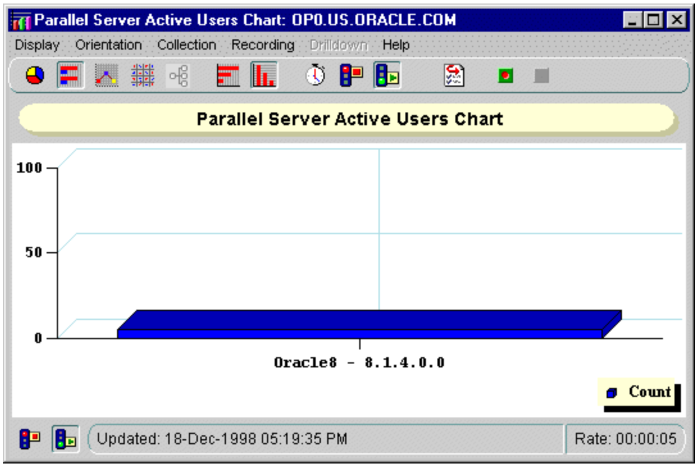
Figure 8–12 Parallel Server Users Logged On by Instance Chart


Parallel Server Users Logged On By Instance Chart: OP0.US.ORACLE.COM
Display
Orientation
Collection
Recording
Drilldown
Help
Parallel Server Users Logged On By Instance Chart
100
50
0
op0
op1
Surr
Updated: 18-Dec-1998 05:12:23 PM
Rate: 00:00:05

Parallel Server Active Users Chart

The Parallel Server Active Users chart displays the total number of user active sessions. This chart displays statistics from the GV$LICENSE view:

Figure 8–13 Parallel Server Active Users Chart


Parallel Server Active Users Chart: OPO.US.ORACLE.COM
Display
Orientation
Collection
Recording
Drilldown
Help
Parallel Server Active Users Chart
100
50
0
Oracle8 - 8.1.4.0.0
Count

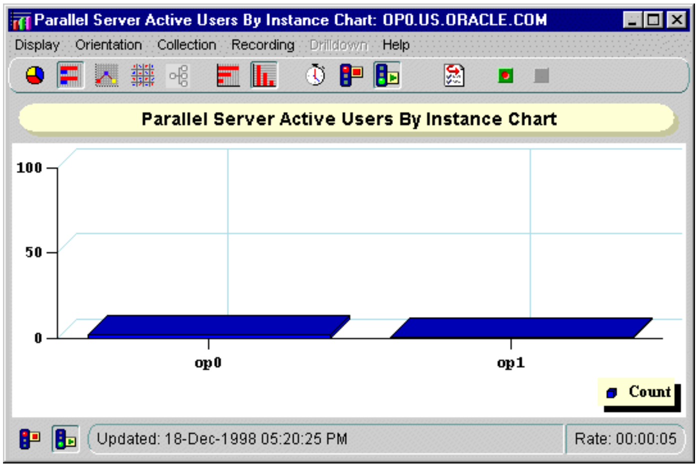
Updated: 18-Dec-1998 05:19:35 PM

Rate: 00:00:05

Parallel Server Active Users by Instance Chart

The Parallel Server Active Users by Instance chart displays the number of active user sessions logged on to each instance in the Oracle Parallel Server. This chart displays statistics from the GV$LICENSE view.

Figure 8–14 Parallel Server Active Users by Instance Chart

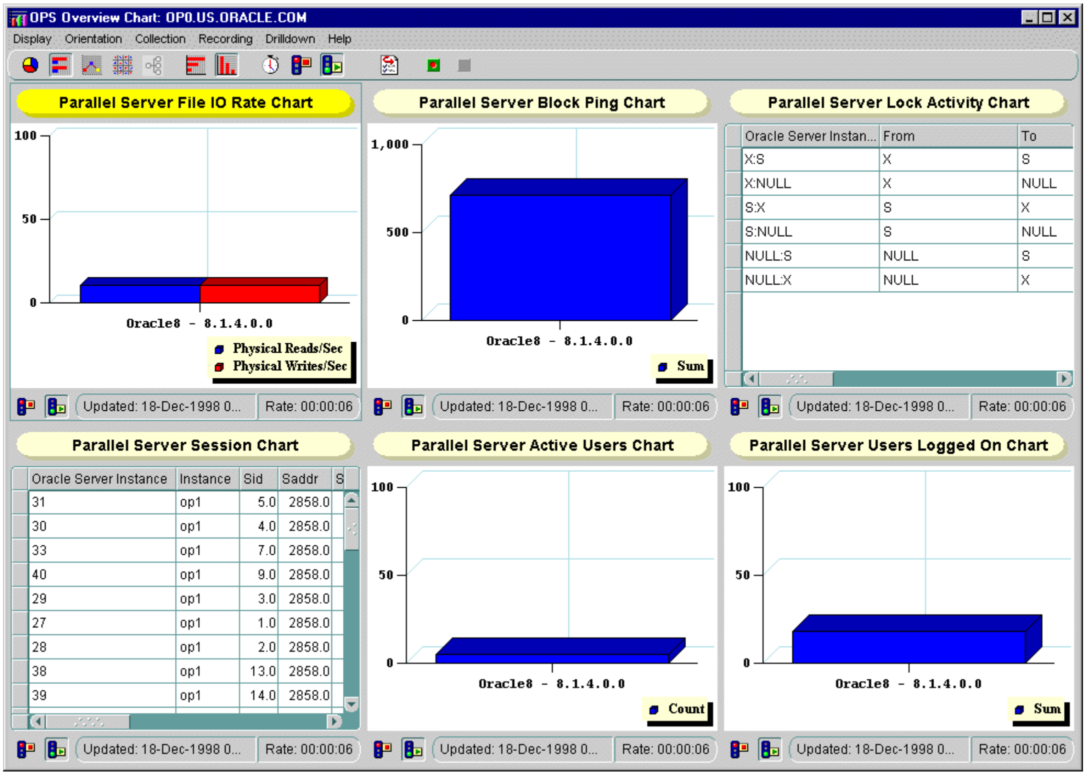

Parallel Server Active Users By Instance Chart: OP0.US.ORACLE.COM
Display
Orientation
Collection
Recording
Drilldown
Help
Parallel Server Active Users By Instance Chart
100
50
0
op0
op1
Count
Updated: 18-Dec-1998 05:20:25 PM
Rate: 00:00:05

OPS Overview Chart

The OPS Overview chart displays the following charts:

- Parallel Server File I/O Rate Chart
- Parallel Server Block Ping Chart
- Parallel Server Lock Activity Chart
- Parallel Server Users Logged On Chart

Figure 8–15 Overview Chart


OPS Overview Chart: OP0.US.ORACLE.COM
Display Orientation Collection Recording Drilldown Help
Parallel Server File IO Rate Chart
100
50
0
Oracle8 - 8.1.4.0.0
Physical Reads/Sec
Physical Writes/Sec
Parallel Server Block Ping Chart
1,000
500
0
Oracle8 - 8.1.4.0.0
Sum
Parallel Server Lock Activity Chart
Parallel Server Session Chart
Parallel Server Active Users Chart
100
50
0
Oracle8 - 8.1.4.0.0
Count
Parallel Server Users Logged On Chart
100
50
0
Oracle8 - 8.1.4.0.0
Sum
Updated: 18-Dec-1998 0...
Rate: 00:00:06

Oracle Server Instan...	From	To
X:S	X	S
X:NULL	X	NULL
S:X	S	X
S:NULL	S	NULL
NULL:S	NULL	S
NULL:X	NULL	X

Oracle Server Instance	Instance	Sid	Saddr	S
31	op1	5.0	2858.0	
30	op1	4.0	2858.0	
33	op1	7.0	2858.0	
40	op1	9.0	2858.0	
29	op1	3.0	2858.0	
27	op1	1.0	2858.0	
28	op1	2.0	2858.0	
38	op1	13.0	2858.0	
39	op1	14.0	2858.0	

Part IV

Adding Instances and Nodes

Part IV describes how to migrate from a single-instance Oracle environment to an Oracle Parallel Server environment. It also explains how to add instances and nodes to an Oracle Parallel Server environment. The chapter in Part IV is:

9

Adding Instances and Nodes

This chapter describes how to convert from a single instance Oracle8*i* database to a multi-instance Oracle8*i* database using the Oracle Parallel Server Option, and how to add nodes to an existing configuration.

Specific topics discussed are:

Moving from a Single Instance to Parallel Server

This section explains how to enable your database structure 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 the multi-instance parallel server. In addition, this chapter can help you extend an existing Oracle Parallel Server configuration to additional nodes. This sections covers the following topics:

- Deciding to Convert
- Preparing to Convert
- Converting the Database from Single- to Multi-instance

Deciding to Convert

You may decide to convert to a multi-instance database when you have designed your application with Oracle Parallel Server in mind.

Do not convert to Oracle Parallel Server in the following situations:

- You are using a file system that is not shared.
- Your application was not designed for parallel processing; you need to examine your application more.
- You are not using a supported configuration (of shared disks, and so on).

Preparing to Convert

This section contains the following sections:

- Hardware and Software Requirements
- Converting the Application from Single- to Multi-Instance
- Administrative Issues

Hardware and Software Requirements

To convert to Oracle Parallel Server you must have:

- A supported hardware and OS software configuration
- A license for the Oracle Parallel Server
- Oracle8*i* Enterprise Edition running on all nodes
- Oracle Parallel Server linked to your system

Converting the Application from Single- to Multi-Instance

Making your database run in parallel does not automatically mean you have effectively implemented Oracle Parallel Server. Besides migrating your existing database from single instance Oracle to multi-instance Oracle, you must also migrate existing applications that were designed for single-instance Oracle. Preparing an application for use with a multi-instance database may require application partitioning and physical schema changes.

> **See Also:** *Oracle8i Parallel Server Administration, Deployment, and Performance* for a full discussion of this topic

Administrative Issues

Note the following administrative issues of conversion:

- Your backup procedures should be in place before converting from single-instance Oracle to the Oracle Parallel Server.
- Additional archiving considerations apply in an Oracle Parallel Server environment. 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, then on systems where file systems cannot be shared, some method of accessing the archive logs is required.

Converting the Database from Single- to Multi-instance

The following procedure explains how to migrate an existing database from single-instance Oracle to multi-instance Oracle. Remember that you must also migrate the application from single- to multi-instance.

To migrate from a single instance to an Oracle Parallel Server, perform the following tasks:

Task 1: Configure Hardware

See your vendor documentation for information about setting up Oracle Parallel Server hardware.

Task 2: Evaluate Tablespaces and Log Files of Single Instance

Because tablespaces must be identical on the Oracle Parallel Server as a single instance database, it is important that you do any consolidation, addition, or renaming of tablespaces on the single instance at this time.

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

Task 3: Create Raw Devices

Create the raw devices needed for the data files, control files, and log files for Oracle Parallel Server, as explained in "Setting Up Raw Devices" on page 2-5.

Task 4: Export Data from Old Database

Export the entire database from the single instance database. Use a tool such as the Export utility

To use the Export utility to export all data from an existing database to the new database:

> **Note:** To export an entire database, you must use the user name SYSTEM. Do not use INTERNAL or SYS.

Run the Export utility:

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

`file``.dmp` represents the data from the full database export of the database, while *`file`*`.log` represents the log of the operation.

> **See Also:** *Oracle8i Utilities* for further information about this tool

Task 5: Install Operating System Dependent Files

For all nodes, see your Operating System Dependent vendor documentation for instructions about installing Operating System Dependent files.

Task 6: Install Oracle8*i* Enterprise Edition and Oracle8*i* Parallel Server

Except on the server already running Oracle8*i* Enterprise Edition, install Oracle8*i* Enterprise Edition along with the Oracle Parallel Server option, as described in Chapter 3. 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 Oracle8*i* Enterprise Edition, install Oracle Parallel Server using the Custom installation type.

Task 7: Create the Database

Create a new database on the raw partitions. Depending on the install type and subsequent configuration options, a database may be created with the following methods:

- If you selected the Typical installation, the database will be created after installation.
- If you requested that Oracle Database Configuration Assistant run after the Custom installation, Oracle Database Configuration creates a database.
- If you requested that Oracle Database Configuration Assistant not run after during Custom installation, you can still run Oracle Database Configuration to create a database, as described in "Use Oracle Database Configuration Assistant in Stand-Alone Mode" on page 3-23.
- If you requested that Oracle Database Configuration Assistant not run during Custom installation, you can create the database manually, as described in "Use Manual Methods" on page 3-29.

Task 8: Transfer Data From Old Data to Empty Database

Import the entire database into the empty database. Use 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*.log represents the log of the operation.

See Also: *Oracle8i Utilities* for further information about this tool

---

ATTENTION: If the original database from which the export file was generated contains tablespaces that are not in the new database, then the Import utility attempts to create those tablespaces with associated data files. The easy solution is to ensure both databases contain the same tablespaces. The data files do not have to be identical. Only the tablespace names need to be the same.

Task 9: Start the Database

See "Starting the Database in Parallel Mode" on page 3-41 to start the Oracle Parallel Server.

Adding Additional Nodes to a Cluster

If you have already configured cluster and want to add other nodes, Oracle recommends the follow procedure:

1. Install another database into a different Oracle home that includes all the nodes you want to support, as described in Chapter 3.
2. Export the data from the existing database, as described in "Task 4: Export Data from Old Database" on page 9-5.
3. Import the data to the new database, as described in "Task 8: Transfer Data From Old Data to Empty Database" on page 9-6.

Part V

Reference

Part V includes appendices with supplemental information about the Oracle Parallel Server installation process. The contents of Part V are:

- Appendix A, "Directory Structure"
- Appendix B, "Oracle Parallel Server Management on UNIX Reference"
- Appendix C, "Troubleshooting"

A

Directory Structure

Specific topics covered in this appendix are:

- Understanding the Oracle Parallel Server Directory Structure
- UNIX
- Windows NT

Understanding the Oracle Parallel Server Directory Structure

When you install Oracle8*i* Enterprise Edition and the Oracle Parallel Server Option, all subdirectories are under a top-level ORACLE_BASE. ORACLE_HOME and `admin` directories are also located under ORACLE_BASE.

UNIX

The following is the hierarchical directory tree of a sample OFA-compliant database for Oracle Parallel Server on UNIX platforms:

$ORACLE_BASE			`/u01/app/oracle` is the default ORACLE_BASE directory
	$ORACLE_HOME		`/product/8.1.6` is the name of the Oracle home by default
	`/bin`		Subtree for Oracle binaries
	`/network`		Subtree for Net8
	`/ops`		Files created by Oracle Database Configuration Assistant
	`/opsm`		Oracle Parallel Server Management message and TCL files
		`/admin`	`ops.sql` script and initialization parameter files for database creation
	`/dbs`		This is a legacy directory from previous releases. It contains links to initialization parameter files that point to the new location files, `$ORACLE_BASE/admin/`*`db_name`*`/pfile`.
	`...`		
`/admin`			Subtree for Oracle Parallel Server 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: *Oracle8i Administrator's Reference* for your UNIX operating system for further information about `$ORACLE_HOME` and `/admin` directories

Windows NT

The following is the hierarchical directory tree of a sample OFA-compliant database for Oracle Parallel Server on Windows NT:

x:\oracle_base				`c:\oracle` is the default ORACLE_BASE directory
	`\%ORACLE_HOME%`			`\ora81` is the name of the Oracle home by default
		`\bin`		Subtree for Oracle binaries
		`\network`		Subtree for Net8 configuration files, including `tnsnames.ora`, `listener.ora` and `sqlnet.ora`
		`\ops`		Files created by Oracle Database Configuration Assistant
		`\opsm`		`\admin` subdirectory
			`\admin`	Oracle Parallel Server `ops.sql` script 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 Oracle Parallel Server 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: *Oracle8i Installation Guide for Windows NT* for further information about the contents of the *ORACLE_HOME* and `\admin` directories

B

Oracle Parallel Server Management on UNIX Reference

This appendix describes Oracle Parallel Server Management on UNIX.

Specific topics covered in this appendix are:

OPSCTL Utility

The OPSCTL utility on UNIX uses a file named *`db_name`*`.conf` to define Oracle Parallel Server instances and related services. The Oracle Database Configuration Assistant create this file during database creation.

> **Important:** The *db_name*.conf file must reside in the Oracle Parallel Server's shared or non-shared Oracle home location, even if this file is empty. OPSCTL requires this file to run properly.

Consider the following points when configuring your *`db_name`*`.conf`:

- The Oracle home location can be set up to be shared or non-shared. If the location is non-shared, you must manually create and remote copy (`rcp`) the *`db_name`*`.conf` file to each node of the cluster.
- The managed system maps the names of the nodes to correspond with non-negative numbers (positive numbers) called "node numbers" based on the thread ID.

Creating the *db_name*.conf File

This section covers the following topics:

- Parameter Syntax
- Parameter Expansion
- Parameter Descriptions
- Parameter Node Overrides
- Oracle Parallel Server Instances

Parameter Syntax

The parameter values are specified in the configuration file using the following syntax:

```
parameter_name = value
parameter_name = (value1, value2, ...)
```

Consider these points when configuring parameter values:

- Enter each parameter on a separate line in the *db_name*.conf file.
- Separate lists of values with commas and enclose them in parentheses.
- You must surround parameter values containing commas or spaces with double quotation marks ("*xxx*").

Parameter Expansion

Parameter values with the percent sign (%) in the configuration file have different meanings depending on the character that follows the % sign:

Value	Description
%p	Expands to the name of the Oracle Parallel Server that is set by the value of ORACLE_PSRV parameter or the environment variable ORACLE_PSRV
%h	Expands to the value of the ORACLE_HOME parameter or expands to the value of the ORACLE_HOME environment variable
%n	Expands to the node number for the appropriate node. The %n is replaced by a different value on each node
%m	Expands to the node name for the appropriate node. The %m is replaced by a different value on each node

Parameter expansion examples are shown in the following table:

Example	Description
`inst_init_ora=/ora/dbs/initop.ora`	The name of the instance-specific parameter file is `initop.ora`. This parameter file applies to all nodes of the Oracle Parallel Server; there is no parameter expansion.

`inst_init_ora=/ora/dbs/initop_ \`
`%m.ora`
	This example, containing `initop_%m.ora`, illustrates parameter expansion which is replaced by a different value on each node. For example, if there are three nodes with the names `spdev01`, `spdev02`, and `spdev03`, the parameter expansion values for these nodes with `node_list = "1-3"` are as follows:
	`initop_spdev01.ora`
	`initop_spdev02.ora`
	`initop_spdev03.ora`

Parameter Descriptions

The following table lists the configuration parameters that you can set in the *db_name*`.conf` file. Each parameter is listed in the order that it would typically appear in the configuration file. You must enter parameters in lowercase.

Parameter	Description
`node_list`	The list of node numbers for all nodes on which Oracle Parallel Server instances are run. This parameter is required on some operating systems, such as IBM RS/6000 SP2, and Sequent. On some operating systems, such as Sun Solaris, the entire cluster is used by default. For these operating systems, this parameter does not need to be explicitly set. Example: `node_list = "1-8"`
`inst_oracle_sid`	*(optional)* The instance identifier. Each Oracle Parallel Server requires a unique Oracle System Identifier (SID). The default is `%p%n`.

Parameter	Description
`inst_init_ora`	*(optional)* A parameter file that is shared by all the instances of the Oracle Parallel Server. If the parameter contains `%n` or `%m`, a different `init`*`sid`*`.ora` on each node is used.
node#`:inst_init_ora`	*(optional)* A parameter file for the Oracle Parallel Server instance on a specific node, as denoted by the *node#*. This overrides any value specified in the `inst_init_ora` parameter.
`oracle_home`	*(optional)* Specifies either the single, shared Oracle home or each non-shared Oracle home location for each node. Defaults to the Oracle home specified in `oratab` (platform-specific location).
`lsnr_listener_name`	*(optional)* The names of the listeners for each specific node.
`tns_admin`	*(optional)* The value of the TNS_ADMIN environment variable that is used to start or stop all listeners.
node#`:tns_admin`	*(optional)* The value of TNS_ADMIN environment variable that is used to start or stop the listener on the specific node, as denoted by *node#*.
`environ`	*(optional)* The double-quoted list of semi-colon separated "*name* = *value*" pairs of environment variables that are required for the startup or shutdown operations. For example: `environ=”ENV1=value1;ENV2=value2”` The environment variables included in this parameter are in addition to ORACLE_PSRV, ORACLE_HOME, and TNS_ADMIN, each of which are represented either by a separate environment variable or a separate configuration parameter.

The following examples show how the configuration parameters are applied. The parameter settings for different services are illustrated with reference to an example Oracle Parallel Server system named, mypsrv, that consists of an eight-node system. The following table shows how node numbers, node names, and instance identifiers are mapped to each other for mypsrv.

Node Number	Node Name	SID
1	spdev01	SID_A
2	spdev02	SID_B
3	spdev03	SID_C
6	spdev06	SID_D
7	spdev07	SID_E
8	spdev08	SID_F

Parameter Node Overrides

You can set the parameters to apply to a specified set of nodes by prefixing the parameter definition with a node number as follows:

node_number:*parameter_name* =*value*

Oracle Parallel Server Instances

Instances are defined by the following parameters:

- inst_oracle_sid
- inst_init_ora

inst_oracle_sid

If the inst_oracle_sid parameter is not specified, then OPSCTL generates a default list of Oracle System Identifiers (SID) in which each SID is set to the following format:

db_name node#

where *db_name* corresponds to the name of the Oracle Parallel Server and where *node#* corresponds to the node number of the corresponding node in the node_ list. This is as if %p%n were specified.

For example, if the parameters are set as follows:

```
oracle_psrv = mypsrv
node_list = ”1-3,6”
```

and you do not specify the inst_oracle_sid parameter, then the SIDs generated for the instances will be mypsrv1, mypsrv2, mypsrv3, and mypsrv6 for the spdev01, spdev02, spdev03, and spdev06 nodes, respectively.

The inst_oracle_sid parameter is used to specify the ORACLE_SIDs (instance identifiers) for the instances. It may be a comma-separated list of SID names where each SID maps to the corresponding node mentioned in the same order in the node_list parameter. Or, it may be that %n or %m is used to generate a distinct SID on each node.

For example, if the parameters are set as follows:

```
node_list = ”1-3,6”
inst_oracle_sid = (sid_a, sid_b, sid_c, sid_d)
```

then the sid_a value will be used for the instance on node spdev01 and the sid_d value will be used for the instance on the spdev06 node.

The parameter substitution facility can be used to customize ORACLE_SID values. For example, if you set the parameters as follows:

```
node_list = ”1-3,6”
inst_oracle_sid = ”%p_SID_%n”
```

then the mypsrv_sid_1, mypsrv_sid_2, mypsrv_sid_3, and mypsrv_sid_6 values are used for SIDs of instances on nodes spdev01, spdev02, spdev03, and spdev06, respectively.

inst_init_ora

The absolute path names for the parameter files are specified by the inst_init_ora and / or *node#*:inst_init_ora parameters. If this parameter is not specified, then $ORACLE_HOME/dbs/init*sid*.ora is used.

For each instance, OPSCTL first searches for the *node#*:inst_init_ora parameter, where *node#* is the number of the node on which the instance runs. If this parameter is set, OPSCTL uses its value as the instance’s parameter file. If it is not specified, then OPSCTL uses the inst_init_ora parameter.

Listeners

Listeners are defined by the following parameters:

- tns_admin
- lsnr_listener_name

tns_admin

The value for tns_admin for all listeners to be started or stopped can be set with the tns_admin parameter.

If the tns_admin parameter is not set and the TNS_ADMIN environment variable is set, the listeners are given the value of the TNS_ADMIN environment variable.

If neither the tns_admin parameter nor the TNS_ADMIN environment variable are set, the listeners are not given any value, and will use the default search order to locate the listener.ora file.

The listener.ora file contains the listening addresses of the listener on the system as well as the name and Oracle home of any database the listener serves.

To use a different TNS_ADMIN variable for a specific node, specify the node using the *node#*:tns_admin parameter, where *node#* refers to the number of the corresponding node.

For example, if the TNS_ADMIN environment variable is set to /home/myname/misc and the tns_admin and none of the *node#*:tns_admin parameters are specified in the configuration file, then OPSCTL uses the value /home/myname/misc for all nodes.

In another example, if the following parameters are set as follows:

```
node_list = "1-3,6"
tns_admin = /dve/myname/815/ops
3:tns_admin = /dve/myname/others
```

then for the listener on spdev03 (node number 3), tns_admin is set to /dve/myname/others, whereas nodes spdev01, spdev02 and spdev06, tns_admin is set to /dve/myname/816/ops.

lsnr_listener_name

The lsnr_listener_name parameter is used to specify the listener names. It is a comma-separated list of listener names, where each listener name maps to the corresponding node mentioned in the same order listed for the node_list parameter.

If you do not specify the lsnr_listener_name parameter, then OPSCTL generates a default list of listener names in which each listener name follows the format *node_name*_listener, where *node_name* corresponds to the name of the node. This is as if %m_listener was specified.

For example, if the node_list parameter is set as follows:

```
node_list = ″1-3,6″
```

and you do not specify the lsnr_listener_name parameter, the names generated for the listeners would be spdev01_listener, spdev02_listener, spdev03_listener, and spdev06_listener for the spdev01, spdev02, spdev03, and spdev06 nodes respectively.

For example, if the parameters are set as follows:

```
node_list = ″1-3,6″
lsnr_listener_name = (listener_A, listener_B, listener_C, listener_D)
```

then the listener_A name would be used for the listener on node spdev01 and the listener_D name would be used for the listener on node spdev06.

If only one entry is specified for the lsnr_listener_name parameter, then the same name is used for listeners on all nodes specified in the node_list. Also, you can use parameter expansion to specify different listener names for all nodes.

For example, if the parameters are set as follows:

```
node_list = ″1-3,6″
lsnr_listener_name = ″listener_mypsrv″
```

then the listener_mypsrv name would be used for all the listeners on all nodes.

In another example, if you set the following parameters as follows:

```
node_list = ″1-3,6″
lsnr_listener_name = ″lsnr_%n″
```

then the names lsnr_1, lsnr_2, lsnr_3, and lsnr_6 will be used for the listeners on the nodes spdev01, spdev02, spdev03, and spdev06, respectively.

Sample of *db_name*.conf File

The following is an example *db_name*.conf file:

```
node_list = "1-4"
inst_init_ora = /ora/dbs/init_%m.ora
inst_oracle_sid = (SID_A, SID_B, SID_C, SID_D)
lsnr_listener_name = listener_myhost%n
```

Oracle Intelligent Agent Configuration

Oracle Intelligent Agents are shipped with the database and must be installed on each remote, managed Oracle Parallel Server node. The Oracle Intelligent Agents are responsible for discovering the available services (databases, listeners, parallel servers) on each node.

The configuration is different for shared and non-shared Oracle home locations. In a typical setup, Oracle Intelligent Agents are installed in non-shared Oracle home locations.

The following directories should *not* be shared because they contain files that are specific to each Oracle Intelligent Agent's node:

- `$ORACLE_HOME/network/admin`
- `$ORACLE_HOME/network/log`
- `$ORACLE_HOME/network/trace`
- `$ORACLE_HOME/network/agent`

CAUTION: An Oracle Intelligent Agent cannot be installed in a shared Oracle home location. You must ensure that each node's Oracle Intelligent Agent is installed in a separate Oracle home location, distinct from other Oracle home locations.

Also ensure that your server files, `tnsnames.ora` and `listener.ora`, contain the correct entries for all instances to perform service discovery, as described in "Oracle Parallel Server Management Requirements" on page 6-6.

Non-shared Oracle Home

You must ensure that you configure and start Oracle Intelligent Agent on every node running an instance.

To start the agent, at an operating system prompt, enter the command:

```
lsnrctl dbsnmp_start
```

Shared Oracle Home

If the Oracle home location is shared (through NFS or any other equivalent facility) by all nodes, you must 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.
5. Ensure that the DBSNMP utility is set up to run at system startup time.

C

Troubleshooting

Specific topics covered in this appendix are:

- Resolving Service Discovery Failures
- Using Trace Files
- Contacting Oracle Worldwide Customer Support

Resolving Service Discovery Failures

Discovery of nodes and objects by Oracle Enterprise Manager typically fails 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 may 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. This file contains information about the nodes and their services (databases, instances and listeners) discovered.

This file is created from the following sources, all on the managed nodes:

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

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

oratab on UNIX and Registry on Windows NT

Discovery first discovers the Oracle Parallel Server database name and the nodes associated with the database. How it accomplishes this depends on whether the managed system is running on:

- UNIX
- Windows NT

UNIX On UNIX operating systems, discovery uses information in the `oratab` entry for the name of the Oracle Parallel Server. `oratab` 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 Oracle Parallel Server database. From this entry, the database name is acquired.

Next, discovery looks for the `node_list` parameter in *db_name*`.conf` file, located in `$ORACLE_HOME/ops`, to determine which nodes run instances of the Oracle Parallel Server:

```
node_list="1,2,3"
```

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.

Windows NT The registry lists all Oracle Parallel Servers that run on a node under the subkey HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\OSD\PM. Under this subkey, each Oracle Parallel Server cluster has its own registry subkey, as described in "Registry Values for Oracle Parallel Server Database on Windows NT" on page 3-10.

Oracle Enterprise Manager uses information to discover the Oracle Parallel Server database name, its instances and nodes. Once the instances and nodes are discovered, the `listener.ora` and `tnsnames.ora` files are inspected.

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 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=opshp1-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=op1)))
  ```

The `listener.ora` file created after installation typically contains the configuration for discovery.

See Also: "Listener (listener.ora file)" on page 3-14

Note: On UNIX operating systems, listeners and instances may also be specified in the *db_name*`.conf` file with the `inst_sid_list` and `lsnr_listener_name` parameters, as described in "Parameter Descriptions" on page B-4.

```
inst_sid_list=(op1, op2)
lsnr_listener_name="listener_%m"
```

where `%m` expands to the node name.

tnsnames.ora

The tnsnames.ora file, located in $ORACLE_HOME/network/admin on UNIX operating systems and *ORACLE_HOME*\network\admin on Windows NT on the discovered nodes, is read by the discovery process to determine names and address information for the Oracle Parallel Server and instances on a node.

Discovery requires the following entries:

- Each instance must have an entry in tnsnames.ora file For example:

```
op1.us.acme.com=
 (description=
  (address=(protocol=tcp)(host=op1-server)(port=1521))
  (connect_data=
    (service_name=op.us.acme.com)
    (instance_name=op1)))
```

- The Oracle Parallel Server should have an entry. For example:

```
op.us.acme.com=
 (description=
  (load_balance=on)
  (failover=on)
  (address_list=
   (address=(protocol=tcp)(host=op1-server)(port=1521))
   (address=(protocol=tcp)(host=op2-server)(port=1521))
  (connect_data=
     (service_name=op.us.acme.com)))
```

- On Windows NT, each instance must have a startup entry named *sid_*startup that uses SID and requests a dedicated server connection in its CONNECT_DATA connection. For example:

```
op1_startup.us.acme.com=
 (description=
  (address=(protocol=tcp)(host=op1-server)(port= 1521))
  (connect_data=
    (sid=op1)
    (server=dedicated)))
```

See Also: "Net Service Names (tnsnames.ora file)" on page 3-16

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.

  The `services.ora` file should contain an `ops_database` entry for the Oracle Parallel Server, that lists the node, database address, and name of the database. The example below shows a database named `op.us.acme.com` running on node `op1-server`. The database address comes from the `op.us.acme.com` net service name in the `tnsnames.ora` file.

  ```
  op.us.acme.com=(ops_database, op1-server, (description=(load_
  balance=on)(failover=on)(address_
  list=(address=(protocol=tcp)(host=op1-server)
  (port=1521))(address=(protocol=tcp)(host=op2-server)(port=1521)))(connect_
  data=(service_name=op.us.acme.com)))), op.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
  - Oracle Parallel Server name to which the instance belongs
  - Listener name

The example below shows the instance OP1 runs on node OPSHP1, and is listened for by LISTENER_OPSHP1:

```
op1.us.acme.com=(ops_instance, op1-server,
(description=(address=(protocol=tcp)(host=op1-server)(port=1521))(connect_
data= (service_name=op.us.acme.com)(instance_name=op1))), op.us.acme.com,
listener_op1-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.

  The following message should be ignored:

  ```
  Warning: No Listener found for sid_db_name, db_name will be
  skipped.
  ```

Troubleshooting Discovery

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

1. Check `nmiconf.lst` in `ORACLE_HOME/network/agent/config` directory on UNIX operating systems and *`ORACLE_HOME`*`\network\agent\config` directory on Windows NT. This file contains a list third-party discovery scripts to run. It must contain at least the following entry:

   ```
   confops.tcl
   ```

   This entry is created during installation of the Oracle Parallel Server.

2. Check that the Oracle Parallel Server is defined correctly:

   On UNIX:

   a. Verify that `oratab` and the *`db_name`*`.conf` file are configured correctly.

   b. Run the following command to verify proper setup:

   ```
   SETENV ORACLE_PSRV db_name
   OPSCTL config -C
   ```

On Windows NT:

a. Check the registry entries associated with the Oracle Parallel Server.

b. Run the following command to verify proper setup:

```
OPSCTL config -ndb_name
```

On UNIX, OPSCTL displays the name of the node, instance, and listener for the node. The following example shows a node named NODE1 running an instance named OP1 with a listener named LISTENER_NODE1.

```
node1 op1 listener_node1
```

On Windows NT, OPSCTL displays a list of all nodes and their corresponding instances. For example:

```
node1 op1
node2 op2
node3 op3
```

3. Inspect the `listener.ora` and `tnsnames.ora` file entries to ensure that the required entries are present.

Using Trace Files

This section discusses the following trace file subjects:

- Background Thread Trace Files
- User Thread Trace Files
- Alert File
- Error Call Trace Stack

Background Thread Trace Files

Oracle Parallel Server background threads use trace files to record occurrences and exceptions of database operations as well as errors. These detailed trace logs are helpful to Oracle support to debug problems in your cluster configuration. Background thread trace files are created regardless of whether the BACKGROUND_DUMP_DEST parameter is set in the `init`*`db_name`*`.ora` initialization parameter file. If you set BACKGROUND_DUMP_DEST, the trace files are stored in the directory specified. If you do not set the parameter, the trace files are stored in:

- `$ORACLE_BASE/admin/`*`db_name/`*`bdump` on UNIX operating systems
- *`ORACLE_BASE`*`\admin\`*`db_name`*`\bdump` on Windows NT

The Oracle8*i* database creates a different trace file for each background thread. The name of the trace file contains the name of the background thread followed by the extension `.trc`, such as:

- *`sid`*`dbwr.trc`
- *`sid`*`smon.trc`

Oracle Parallel Server trace information is reported in the following trace files:

Trace File	Description
`sid``bsp0.trc`	Trace file for Cache Fusion BSP (Block Server Process). This trace files shows errors associated with BSP.
`sid``lckn.trc`	Trace file for the LCK*n* process. This trace file shows lock requests for other background processes.
`sid``lmdn.trc`	Trace file for the LMD*n* process. This trace file shows lock requests.
`sid``lmon.trc`	Trace file for the LMON process. This trace file shows the status of the cluster.
`sid``p00n.trc`	Trace file for the parallel execution processes.

User Thread Trace Files

Trace files are also created for user threads if the USER_DUMP_DEST parameter is set in the initialization parameter file. The trace files for the user threads have the form `ora`*`xxxxx`*`.trc`, where *xxxxx* is a 5-digit number indicating the process ID on UNIX or the Windows NT thread ID.

Alert File

The alert file, *`sid`*`alrt.log`, contains important information about error messages and exceptions that occur during database operations. Each instance has one alert file; information is appended to the file each time you start the instance. All threads can write to the alert file.

`sid``alrt.log` is found in the directory specified by the BACKGROUND_DUMP_DEST parameter in the `init`*`db_name`*`.ora` initialization parameter file. If you do not set the BACKGROUND_DUMP_DEST parameter, the *`sid`*`alrt.log` file is generated in:

- `$ORACLE_BASE/admin/`*`db_name`*`/bdump` on UNIX operating systems
- *`ORACLE_BASE`*`\admin\`*`db_name`*`\bdump` on Windows NT

Error Call Trace Stack

Oracle Worldwide Support may ask you to create an error call trace stack for a particular trace file. An error call trace stack provides a program trace of specific background or user threads in the database.

To create an error call trace:

1. Obtain the Oracle process ID for the background processes:

   ```
   SQL> CONNECT internal/password
   SELECT pid "Oracle Process Id",
          name
       from v$process, v$bgprocess
       where v$process.addr = v$bgprocess.paddr;
   ```

   The output displayed should resemble the following:

   ```
   Oracle Pro NAME
   ---------- -----
            2 PMON
            3 LMON
            4 LMD0
            5 DBW0
            6 LGWR
            7 CKPT
            8 SMON
            9 RECO
           10 SNP0
           11 SNP1
           13 LCK0
   ```

2. Dump the trace stack to the trace file. For example, to dump the trace stack of LMON, enter:

   a. Set the Oracle process ID to LMON, which is 3 in this example:

      ```
      ORADEBUG setorapid 3
      ```

   b. Dump the error stack to *sid*`lmon.trc`:

      ```
      ORADEBUG dump errorstack 3
      ```

Contacting Oracle Worldwide Customer Support

If after reading this appendix, you still cannot resolve your problems, call Oracle Worldwide Customer Support to report the error. Please have the following information available:

- Your cluster hardware specifications, for example, a two-node cluster of Dell PowerEdge 6100 servers
- Your operating system version
- All five digits of the release number of your Oracle RDBMS (for example, 8.1.6.0.0)
- All five digits in the release number of your Oracle Parallel Server
- On Windows NT, the contents of HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\OSD key
- Cluster OSD upgrades from vendor
- Information about the particular operation that failed, for example, database startup or query
- A list of steps that reproduce the problem.

Severe Errors

If an ORA-600 error occurred, it will be printed to `sidalrt.log` file. If an ORA-600 error or any other severe errors appear in the `sidalrt.log` file, then provide all files in:

- `$ORACLE_BASE/admin/db_name/bdump` on UNIX operating systems
- `ORACLE_BASE\admin\db_name\bdump` on Windows NT

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

alert file

A file that contains important information about error messages and exceptions that occur during database operations.

BSP0 process

A process that manages out-going messages to requesting nodes for Cache Fusion.

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 Oracle8*i* release 8.1 database instance. The **LMON process**, a background process that monitors the health of the **Distributed Lock Manager (DLM)**, 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 database 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 **MTS** 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.

data file

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

dedicated server

A server that requires a dedicated server process for each user process. There is one server process for each client. Net8 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 **multi-threaded server (MTS)**.

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 **multi-threaded server (MTS)**.

Distributed Lock Manager (DLM)

Oracle Parallel Server software that provides locking mechanisms to control allocation and modification of Oracle resources.

extended partition

A type of partition on Windows NT that points to raw space on the disk. An extended partition can be assigned multiple logical drives to accommodate data files, 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.

global cache

Initialization parameters used to determine the size of the collection of global locks that protect the database buffers on all instances. These parameters should be set in the `init`*`db_name`*`.ora` file.

global database name

The full name of the database which uniquely identifies it 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, `sales`, is a simple name 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 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, `init`*`db_name`*`.ora`.

HOME_NAME

The name of an Oracle home in an **Oracle services** on Windows NT.

initialization parameter file

Files that contains information to initialize the database (`init`*`db_name`*`.ora`) and instances (`init`*`sid`*`.ora`).

init*sid*.ora

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

init*db_name*.ora

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

Input/Output (IO)

Input/Output is an Operating System Dependent component that provides I/O to access shared disks.

instance

For an Oracle Parallel Server database, each node within the cluster has an instance of the running Oracle8*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 parallel server database.

Each instance has 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, `init`*`sid`*`.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 allows 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, `init`*`sid`*`.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.

LCK*N* process

A process that manages the locks used by an instance and coordinates requests for those locks by other instances. Additional lock processes, LCK1 through LCK9, are available for systems requiring exceptionally high throughput of instance lock requests. The single lock process per instance, LCK0, is usually sufficient for most systems.

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.

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

LMD*N* process

A process that handles remote lock requests (those which originate from another instance).

LMON process

A process that manages instance and process deaths and associated recovery for the **Distributed Lock Manager (DLM)**.

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)

A server that is configured to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With MTS configuration, 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 with **dedicated server**.

MTS

See **multi-threaded server (MTS)**.

multiple Oracle homes

The capability of having more than one Oracle home directory on a machine.

naming method

The method used by a client application 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
- External naming service, such as Novell Directory Services (NDS), Network Information Service, or Cell Directory Service (CDS)

Net8

The foundation of Oracle's family of networking products. It allows services and their applications to reside on different computers so they can communicate as peer applications. The main function of Net8 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, Net8 acts as a data courier for the client and the server.

Net8 Assistant

Net8 Assistant is a graphical user interface tool that combines configuration abilities with component control to provide an integrated environment for configuring and managing Net8. It can be used on either the client or server.

You can use Net8 Assistant 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. The Net8 Assistant supports 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.

Net8 Configuration Assistant

A post-installation tool that configure 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

node

A machine where an instance resides.

Operating System Dependent layer

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 Oracle Parallel Server.

OPSCTL utility

A utility that manage instances. OPSCTL gathers information about all the instances for **Oracle Enterprise Manager**. OPSCTL 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 OPSCTL. OPSCTL on that node then communicates to the other nodes through **Net8**.

OPS Communication Daemon (OPSD)

A process that receives requests from the **OPSCTL utility** to execute administrative job tasks, such as startup or shutdown. The command is executed locally on each node and the results are returned to OPSCTL. OPSD is installed on the nodes. OPSD is only implemented on UNIX platforms.

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

A tool that enables you to create, delete, and modify a database.

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 via the **Management Server**. High availability is ensured since the agent can function regardless of the status of the **Console** or network connections.

Oracle Parallel Execution

Divides the work of processing certain types of SQL statements among multiple parallel execution server processes.

Oracle Parallel Server

An architecture that allows multiple instances to access a shared database of data files. Oracle Parallel Server is also a software component that provides the necessary Oracle Parallel Server scripts, initialization files, and data files to make the Oracle8*i* Enterprise Edition an Oracle Parallel Server database.

Oracle Parallel Server Management

A comprehensive and integrated system management solution for the Oracle Parallel Server. Oracle Parallel Server Management allows you to manage multi-instance databases running in heterogeneous environments through an open client-server architecture through **Oracle Enterprise Manager**.

In addition to managing parallel databases, Oracle Parallel Server Management allows you to schedule jobs, perform event management, monitor performance, and obtain statistics to tune parallel databases.

Oracle Performance Manager

An add-on application for **Oracle Enterprise Manager** that offers a variety of tabular and graphic performance statistics for parallel servers. The statistics represent the aggregate performance for all instances running on an Oracle Parallel Server.

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 Oracle Parallel Server 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 was used to identify 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.

Oracle8*i* Enterprise Edition

Oracle8*i* Enterprise Edition is an Object-Relational Database Management System (ORDBMS). It provides the applications and files to manage a database. All other Oracle Parallel Server components are layered on top of Oracle8*i* Enterprise Edition.

PMON process

A process monitor database process that performs process recovery when a user process fails. PMON is responsible for cleaning up the cache and freeing resources that the process was using. PMON 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**.

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.

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 RBS*thread_id_rollback_number* by the ROLLBACK_SEGMENTS parameter in the instance initialization file, `init`*`sid`*`.ora`.

seed database

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

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, `init`*`dbname`*`.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.

SID

SID is an abbreviation for **Oracle System Identifier (SID)**.

sqlnet.ora file

A configuration file for the client or server that specifies:

- Client domain to append to unqualified service names or net service names
- Order of naming methods the client should use when resolving a name
- Logging and tracing features to use

- Route of connections
- Preferred Oracle Names servers
- External naming parameters
- Oracle Advanced Security parameters

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

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 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 data files. 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, `init`*`sid`*`.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 file

A configuration file that contains **net service names** mapped to **connect descriptors**. The `tnsnames.ora` file typically resides in `$ORACLE_HOME/network/admin` on UNIX platforms and *`ORACLE_HOME`*`\network\admin`.

This file is needed on clients, nodes, the **Console**, and on the **Oracle Performance Manager** machine.

Transparent Application Failover (TAF)

A runtime failover for high-availability environments, such as Oracle Parallel Server and Oracle Fail Safe, 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.

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.

Index