Oracle® Parallel Server

Getting Started

Release 8.0.6 for Windows NT

June 1999

Part No. A69942-01

ORACLE

Oracle Parallel Server Getting Started, Release 8.0.6 for Windows NT

Part No. A69942-01

Contributors: Eric Belden, Helen Slattery, Deborah Steiner, Database Server Division Engineering

Contents

7 Administering Multiple Instances

8 Adding Instances and Nodes

Contact Us!

Oracle Parallel Server Getting Started, Release 8.0.6 for Windows NT

Part No. A69942-01

This document describes how to contact Oracle Corporation if you have issues with the documentation or software.

Read the section...	If you...
How to Contact Oracle Technical Publications	Have issues with Documentation
How to Contact Oracle Support Services	Have issues with Software
Resources for Oracle Partners and Developers	Want to join an Oracle partner or application developer program

How to Contact Oracle Technical Publications

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?
- Do you have suggestions for improvement? Please indicate the chapter, section, and page number (if available).

You can send comments regarding documentation to *ntdoc@us.oracle.com*

How to Contact Oracle Support Services

Please copy this page and distribute within your organization as necessary.

Oracle Support Services can be reached at the following telephone numbers. The hours of business are detailed in your support contract and the *Oracle Customer Support Guide* in your kit.

Oracle Support Services In...	Call...
All locations	The telephone number for your country is listed at the following Web site: `http://www.oracle.com/support/contact_us/sup_hot_phone.html` Oracle Support Services telephone numbers are also listed in the *Oracle Customer Support Guide* in your kit.

Please complete the following checklist before you call. If you have this information ready, your call can be processed much quicker.

❑ Your CPU Support Identification Number (CSI Number) if applicable.

_____

_____

❑ The hardware name on which your application is running.

_____

_____

_____

❑ The operating system name and release number on which your application is running.

_____

_____

_____

- ❑ The release numbers of the Oracle Server and associated products involved in the current problem. For example, Oracle8 Server release 8.0.6.0.0 and Oracle Replication Manager release 1.6.0.0.0.

- ❑ Are you using 16-bit or 32-bit Oracle software?

- ❑ The third-party vendor and version you are using.

- ❑ The exact error codes and messages. Please write these down as they occur. They are critical in helping Oracle Support Services to quickly resolve your problem.

- ❑ A description of the issue, including:
  - **What happened?** For example, the command used and its result.

- **When did it happen?** For example, during peak system load, or after a certain command, or after an operating system upgrade.

  _____

  _____

  _____

- **Where did it happen?** For example, on a particular system, or within a certain procedure or table.

  _____

  _____

  _____

- **What is the extent of the problem?** For example, production system unavailable, or moderate impact but increasing with time, or minimal impact and stable.

  _____

  _____

  _____

❑ Keep copies of any trace files, core dumps, and redo log files recorded at or near the time of the incident. Oracle Support Services may need these to further investigate your problem.

Resources for Oracle Partners and Developers

This section provides information on partner programs and resources for Oracle database administrators and application developers.

Information Source	Description
Oracle Corporation Home Page `http://www.oracle.com`	This Web site is the starting point for general information on Oracle Corporation.
Alliance Online `http://alliance.oracle.com`	Oracle provides leading-edge technology, education, and technical support that enables you to effectively integrate Oracle into your business. By joining the Oracle Partner Program, you demonstrate to customers that you are committed to delivering innovative Oracle-based solutions and services. The greater your commitment to Oracle, the more we can help you grow your business. It's that simple. The value you derive is associated directly with your level of commitment.
Oracle Education `http://education.oracle.com/`	Customers come to Oracle Education with a variety of needs. You may require a complete curriculum based on your job role to enable you to implement new technology. Or you may seek an understanding of technology related to your key area of responsibility to help you meet technical challenges. You may be looking for self-paced training that can be used as an ongoing resource for reference and hands-on practice. Or, you may be interested in an overview of a new product upgrade. Whatever your training need, Oracle Education has the solution.
Oracle Technology Network `http://technet.oracle.com/`	The Oracle Technology Network is your definitive source for Oracle technical information for developing for the Internet platform. You will be part of an online community with access to free software, Oracle Technology Network-sponsored Internet developer conferences, and discussion groups on up-to-date Oracle technology. Membership is free.
Oracle Store `http://oraclestore.oracle.com/`	This is Oracle's online shopping center. Come to this site to find special deals on Oracle software, documentation, publications, computer-based training products, and much more.

Information Source	Description
Oracle Support Services' Support Web Center `http://www.oracle.com/support/`	Oracle Support Services offers a range of programs so you can select the support services you need and access them in the way you prefer: by telephone, electronically, or face to face. These award-winning programs help you maintain your investment in Oracle technology and expertise.
Oracle*MetaLink* `http://www.oracle.com/support/elec_sup/index.html`	Oracle*MetaLink* is Oracle Support Services' premier Web support service. It is available to Oracle*metals* customers (Gold, Silver, Bronze), 24 hours a day, seven days a week.
Oracle*Lifecycle* `http://www.oracle.com/support/sup_serv/lifecycle/index.html`	Oracle*Lifecycle* is designed to deliver customized, industry-focused, full life-cycle support solutions that enable industry leaders to use Oracle technology to make smart business decisions, achieve operational excellence, and succeed in their markets.
Expert*ONLINE* `http://www.oracle.com/support/sup_serv/online/index.html`	Oracle Support Services has launched a new line of services called Expert*ONLINE*. These services provide online database administration for companies looking to supplement their existing DBA staff or fill a DBA role. Services range from Expert*DETECT*, a monitoring, diagnostic, and recommendation service, to Expert*DBA*, a full online database administration service.
Virtual Support Analyst (VSA) `http://www.oracle.com/support/sup_serv/vsa_start.html`	VSA is Oracle's Internet e-mail service; it is available to U.S. customers with an Oracle*metals* support agreement. With VSA, you can initiate a request for assistance through e-mail, bypassing the queues you may encounter when using telephone support. VSA also enables you to access Oracle's bug database.
Customer Service `http://www.oracle.com/support/cus_serv/index.html`	This site provides resources to make your interactions with Oracle as easy as possible. Among the things you can do are ■ Learn what is a CPU Support Identification (CSI) number ■ Update your technical contact information ■ Find out whom to contact for invoice and collection issues ■ Request product update shipments ■ Access a glossary of Oracle Support Services terms

Information Source	Description
U.S. Customer Visit Program `http://www.oracle.com/support/cus_serv/cus_visit.html`	This U.S.-based program has been established to help our customers understand and obtain maximum benefit from the support services they have purchased. The visit typically offers a customized orientation presentation, a comprehensive overview and demonstration of Oracle's electronic services, and helpful tips on working more effectively with Oracle Support Services.
Support Web Center Library `http://www.oracle.com/support/library/index.html`	This site contains articles, guides, and other documentation to help you leverage the wealth of knowledge and reference material that Oracle Support Services produces.
Product Availability *infowin@us.oracle.com*	Send an e-mail to request information on future product releases on Oracle for Windows NT and Windows 95/98.

Preface

This guide provides operating system-specific information on using Oracle Parallel Server for Windows NT. Topics discussed are:

- Intended Audience
- Prerequisites
- Structure
- Conventions
- Types of Documentation
- Related Documents

Intended Audience

This guide is for network administrators and database administrators who install, configure, and use Oracle Parallel Server on Windows NT clusters.

Prerequisites

This guide assumes that you are familiar with:

- Windows NT and have installed and tested it for your personal computer (PC) and network hardware
- object relational database management concepts

Structure

This guide contains the following chapters and appendices.

Chapter 1, "Overview of Oracle Parallel Server"

Describes the main features and components of Oracle Parallel Server.

Chapter 2, "System Requirements"

Describes the specific hardware and software requirements for installing Oracle Parallel Server.

Chapter 3, "Performing Pre-Installation Tasks"

Guides you through necessary pre-installation tasks

Chapter 4, "Installing Oracle Parallel Server"

Describes how to install how to install the components needed for Oracle Parallel Server.

Chapter 5, "Configuring Oracle Parallel Server"

Describes how to configure an Oracle Parallel Server.

Chapter 6, "Installing and Configuring Oracle Parallel Server Manager"

Describes the tasks to install and configure Oracle Enterprise Manager and Performance Manager for parallel server management.

Chapter 7, "Administering Multiple Instances"

Describes Oracle Parallel Server initialization files and Oracle Parallel Server Manager.

Chapter 8, "Adding Instances and Nodes"

Describes how to add a third or fourth node to an existing cluster, and how to migrate from an single-instance database to multiple instances.

Chapter 9, "Backing Up and Recovering"

Describes NT Backup Manager and the NT Recovery Manager.

Appendix A, "Directory Structure"

Describes the directory structures for Oracle Parallel Server.

Appendix B, "Troubleshooting"

Describes how to resolve Oracle Parallel Server problems.

Glossary

Provides brief descriptions of terms used throughout this manual.

Conventions

The following conventions are used in this guide:

Convention	Example	Meaning
All uppercase plain	ORANT\DATABASE\INITORCL.ORA	Indicates command names, SQL reserved words, and keywords, as in ALTER DATABASE. All uppercase plain is also used for directory names and file names
Italic	Italic used to indicate a variable: *filename*.ORA Italic used for emphasis: The WHERE clause may be used to *join* rows in different tables.	Indicates a value that you must provide. For example, if a command asks you to type *filename*, you must type the actual name of the file. Italic is also used for emphasis in the text and to indicate the titles of other guides.
Square brackets [ ]	Start > Programs > Oracle for Windows NT - [*HOME_NAME*] > Oracle Database Assistant	Encloses optional items. For example, you can start Oracle Installer from the default Oracle home or from another Oracle home indicated by [*HOME_NAME*] if you use multiple Oracle homes. Square brackets also indicate a function key, for example [Enter].
Choose Start >	Choose Start > Programs > Oracle for Windows NT > Oracle Installer	How to start a program. For example, to start Oracle Installer, you must click the Start button on the taskbar and then choose Programs, Oracle for Windows NT, Oracle Installer.
C:\>	C:\ORANT\DATABASE	Represents the Windows NT command prompt of the current hard disk drive. Your prompt reflects the subdirectory in which you are working. Referred to as the *MS-DOS command prompt* in this guide.
Backslash (\) before a directory name	\DATABASE	Indicates that the directory is a subdirectory of the root directory.

Convention	Example	Meaning
ORACLE_HOME	Go to the *ORACLE_HOME*\DATABASE directory	*ORACLE_HOME* is represented as the hard drive letter and the top level directory where the Oracle software is installed, such as: C:\ORANT for Windows NT C:\ORAWIN95 for Windows 95 C:\ORAWIN98 for Windows 98
	`SVRMGR> @%ORACLE_HOME%\RDBMS80\ADMIN\CATALOG.SQL`	In Server Manager commands, you may see %ORACLE_HOME%. Server Manager is able to locate your Oracle Home directory using the %ORACLE_HOME% variable. This convention can be used in Server Manager, SQL*Plus, Export Utility, and Import Utility.
HOME_NAME	Oracle*HOME_NAME*TNSListener80	Represents the Oracle home name if you use multiple Oracle homes. *This convention is not applicable for a single Oracle home.* The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.
HOME*ID*	HOME0, HOME1, HOME2	Represents a unique registry subkey for each Oracle home directory in which you install products. A new HOME*ID* is created and incremented each time you install products to a different Oracle home directory on one computer. Each HOME*ID* contains its own configuration parameter settings for installed Oracle products.
Symbols	period . comma , hyphen - semicolon ; colon : equal sign = backslash \ single quote ‘ double quote “ parentheses ()	Symbols other than brackets and vertical bars must be entered in commands exactly as shown.

Types of Documentation

Your documentation set consists of two types of documentation:

Documentation Type	Describes...
Operating System-specific	Installation, configuration, and use of Oracle Parallel Server in a Windows NTenvironment. Operating system-specific documents are occasionally referred to in the generic documentation set. These documents are easy to identify because they always mention their specific operating system in their title.
Generic	Products that are uniform across all operating system platforms. The vast majority of documents in your documentation set belong to this category. While reading through the generic documentation set, you are occasionally asked to refer to your platform-specific or operating system-specific documentation for procedures specific to the Windows NT operating systems. To easily identify where these generic documentation references are described in this document, see the index of this guide for the following entry: generic documentation references All generic documentation references described in this guide appear under this index entry.

Related Documents

For more information, see the following user guides:

- *Oracle8 Enterprise Edition for Windows NT Installation CD-ROM Insert*
- *Oracle8 Enterprise Edition Release Notes*
- *Oracle8 Enterprise Edition Getting Started for Windows NT*
- *Oracle Enterprise Manager Configuration Guide*
- *Oracle Enterprise Manager Administrator's Guide*
- *Net8 Getting Started for Windows NT and Windows 95*
- *Net8 Administrator's Guide*
- *Oracle8 Parallel Server Concepts and Administration*
- *Oracle Parallel Server Management User's Guide*
- *Oracle8 Concepts*
- *Oracle8 Reference*
- *Oracle8 Error Messages*

1

Overview of Oracle Parallel Server

This chapter provides a conceptual and component overview of Oracle Parallel Server. This information helps you prepare and plan your Oracle Parallel Server installation. Specific topics discussed are:

- Introduction
- Oracle Parallel Server Software Components
- Oracle Parallel Server Instances
- Oracle Parallel Server Features

Introduction

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

- 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

An Oracle Parallel Server can handle node or disk failure in a clustered environment with no or minimal downtime. The Oracle Parallel Server architecture provides the following features:

- Separate Oracle instances running on different nodes operate against a set of common database files that reside on shared disks physically accessible by all nodes that make up the cluster.
- All instances can execute transactions concurrently against the same database, and each instance can have multiple users executing transactions concurrently.

Coordination of each node accessing the shared database provides the following:

- Data consistency and integrity
- Data and lock recovery support

Oracle Parallel Server Software Components

The following components make up an Oracle Parallel Server:

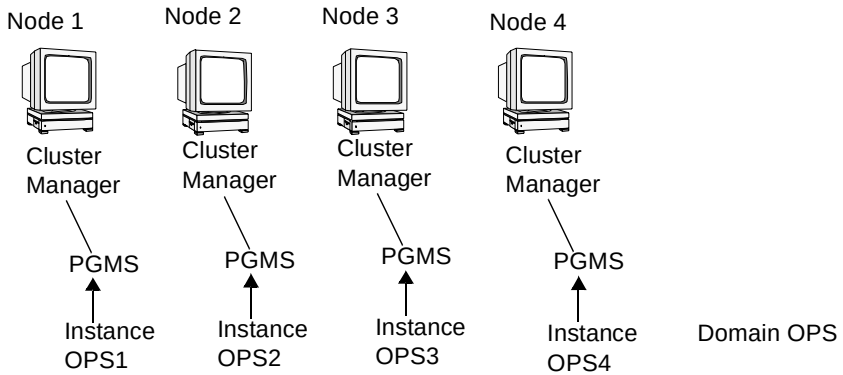
Component	Description
Oracle8 Enterprise Edition	Provides the applications and files to manage a database. All other Oracle Parallel Server components are layered on top of Oracle8 Enterprise Edition.
Oracle Parallel Server Option Server	Provides the necessary Oracle Parallel Server scripts and files to create and support an Oracle Parallel Server. This option also installs the Group Membership Service.
Oracle Parallel Server Manager	Provides a single point for starting, stopping, and monitoring the activity of parallel servers and parallel server instances from the command line or from within Oracle Enterprise Manager Console. **Additional Information**: See Chapter 6, "Installing and Configuring Oracle Parallel Server Manager", and "Managing Instances Using OPSM" in Chapter 7, "Administering Multiple Instances".

Component	Description
Group Membership Service (PGMS)	Monitors instances on nodes. PGMS is installed as a part of the Oracle Parallel Server Option. **Additional Information:** See the "Group Membership Service" on page 1-3.
Operating System Dependent (OSD) layer	Consists of several software components developed by vendors. The OSD layer maps the key OS/cluster-ware services required for proper operation of Oracle Parallel Server. **Additional Information:** See the "Operating System Dependent Layer" on page 1-4.

Group Membership Service

The Group Membership Service (PGMS) called OraclePGMSService on each node monitors what groups or domains are up and its instance members. OraclePGMSService interacts with the Cluster Manager — the vendor software that manages access to shared disks and monitors the status of various cluster resources, including nodes, networks, and the PGMS. The Cluster Manager provides a node monitor service, described in "Cluster Manager" on page 1-5, which each PGMS uses to determine the status of PGMS instances on other nodes. Each instance attaches to the OraclePGMSService during instance startup. An instance detaches from the OraclePGMSService during shutdown. Figure 1–1, "PGMS" depicts how PGMS coordinates between Cluster Manager and an instance.

Figure 1–1 PGMS



Operating System Dependent Layer

A vendor-supplied Operating System Dependent (OSD) layer that passed certification must be installed after Oracle Parallel Server Option is installed. The OSD layer consists of several software components developed by vendors. The OSD layer maps the key OS/cluster-ware services required for proper operation of Oracle Parallel Server.

OSD Components

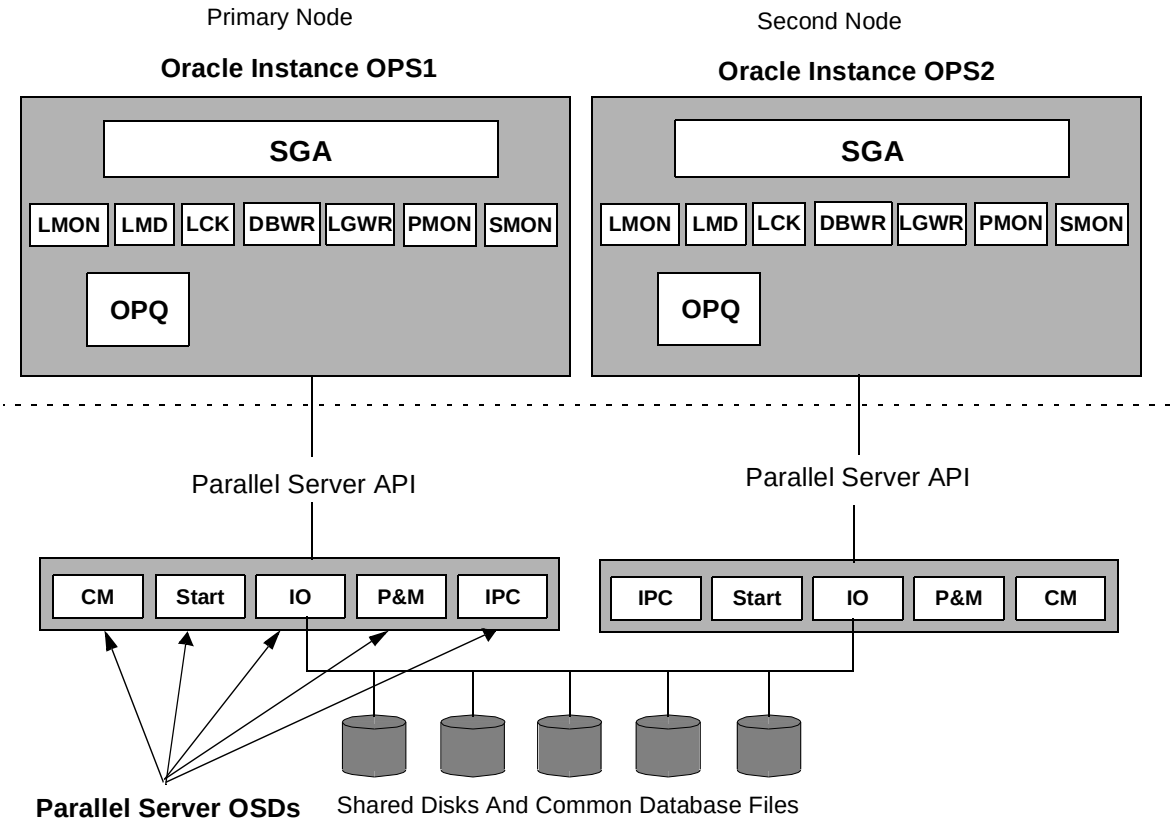
The OSD layer consists of:

Component	Description
Cluster Manager (CM)	Discovers the state of the cluster.
Inter-Process Communication (IPC)	Provides reliable transfer of messages between instances on different nodes.
Input/Output (IO)	Provides I/O to access shared disks.
Startup (START)	Provides one-time configuration to startup functionality.
Performance and Management (P&M)	Supports external performance and management tools.

These components provide key services required for proper operation of the Oracle Parallel Server Option and are used by various clients, such as PGMS and Integrated Distributed Lock Manager (IDLM). Each OSD module interacts with the Oracle Parallel Server runtime environment as a single Dynamic Link Library (DLL). These components are more fully described later in this chapter.

Figure 1–2, "OSD Components" illustrates the OSD components in a cluster with two nodes.

Figure 1–2 OSD Components



Cluster Manager

The Cluster Manager (CM) component:

- Provides the basic node management interface modules needed by Oracle Parallel Server in a cluster environment.
- Allows the Oracle Parallel Server environment to discover and track the membership state of nodes by providing a common view of cluster membership across the cluster.

- Runs on all nodes and monitors the topology of the cluster by querying all nodes for their current membership. Note this is different than PGMS, where monitoring occurs at the instance level rather than the node level.
- Detects changes in the state of active nodes signaling those events, diagnosing the changes, and coordinating a new common and consistent state among all nodes.
- Notifies Oracle Parallel Server of the actual changes to membership.

It is critical that all Oracle Parallel Server instances receive the same membership information when events occur. Notification changes cause relevant Oracle Parallel Server recovery operations to be initiated. If any node is determined to be dead or otherwise not a properly functioning part of the system, CM terminates all processes on that node. Thus, any process or thread running Oracle code can safely assume its node is an active member of the system.

If there is a failure, recovery is transparent to user applications. CM automatically reconfigures the system to isolate the failed node and notify PGMS of the status. PGMS then notifies the IDLM. The IDLM subsequently recovers any of the locks from the failed node. Oracle Parallel Server can then recover the database to a valid state.

The IDLM relies on the Cluster Manager for timely and correct information. If the IDLM cannot get the information it needs, it will shut down the instance.

Inter-Process Communication (Node-to-Node)

Oracle Parallel Server derives most of its functional benefits from its ability to run on multiple interconnected machines. Oracle Parallel Server relies heavily on the underlying Inter-Process Communication (IPC) component to facilitate this.

IPC defines the protocols and interfaces required for the Oracle Parallel Server environment to transfer reliable messages between instances. Messages are the fundamental logical units of communication in this interface. The core IPC functionality is built around an asynchronous, queued messaging model. IPC is designed to send/receive discrete messages as fast as the hardware allows. With an optimized communication layer, various services can be implemented above it. This is how the IDLM carries out all of its communication.

Input/Output

The Input/Output (IO) component provides interprocess capabilities that a cluster implementation must support to enable proper operation of the Oracle Parallel Server environment.

The Oracle Parallel Server environment is extremely dependent on the ability of the underlying OS/cluster implementation to support simultaneous disk sharing across all nodes that run coordinated Oracle Parallel Server instances. Unlike switch-over based technologies, all Oracle Parallel Server instances are active and can operate on any database entity in the shared physical database simultaneously. It is this capability that gives Oracle Parallel Server a large portion of its parallel scalability. It is the role of the IDLM to coordinate the simultaneous access to shared databases in a way that maintains consistency and data integrity.

At a high level, the Oracle Parallel Server shared I/O model can be described as a distributed disk cache implemented across all nodes that define the Oracle Parallel Server cluster. The core of Oracle Parallel Server can be viewed as a major client of the cache. Disk blocks from the shared devices are read into a particular node instance cache only after mediation by the IDLM. The other node instance may read the same blocks into its cache and operate on them simultaneously. Updates to those blocks are carefully coordinated. In general, all shared disk based I/O operations are mediated by the IDLM. The set of distributed IDLMs on each node can be thought of as managing the distributed aspects of the cache.

Disk update operations must be carefully coordinated so that all nodes see the same data in a consistent way. Any Oracle Parallel Server instance intending to update a cached data block must enter into a dialog with the IDLM to ensure it has exclusive right to update the block. Once it does this, the instance is free to update the block until its rights have been revoked by the IDLM. When the exclusive update right is revoked, the instance with block updates must write the block to disk so that the other node can see the changes. Given this rather high-level view of the IDLM I/O consistency model, it is clear that disk blocks can migrate around to each instance's block cache and all updates are flushed to disk when an instance other than the owner desires access to the block. It is this property that directly determines the reliance of Oracle Parallel Server on shared disk implementations.

Startup

The Startup (START) component initiates the Oracle Parallel Server components in a specific order during instance startup. It is up to the vendor to determine this startup sequence.

Performance and Management

The Performance and Management (P&M) component defines the way Oracle Parallel Server clients are configured and the relevant information an OSD implementation must provide to allow the Oracle management tools to operate

properly. The P&M component also defines the way OSD modules provide performance analysis information suitable for OPSM.

Oracle Parallel Server Instances

Oracle Parallel Server instances coordinate with the following components:

Component	Description
node	A Windows NT server where an instance resides.
cluster	A set of physically interconnected nodes, and a shared disk storage subsystem.
domain	The set of all instances coordinating together. A domain is limited to a set of Oracle Parallel Server instances that run on only the nodes defined within a cluster. The domain is defined by the DB_NAME in the INIT_COM.ORA file.

Oracle Parallel Server allows multiple instances to coordinate data manipulation operations on a common database. An Oracle Parallel Server *instance* is defined as a process and a set of threads and memory structures required for proper database operation. The pool of threads that make up an instance coordinate their work on the database through shared memory by means of the instance Shared Global Area (SGA). Users can connect to any instance to access the information that resides within the shared database.

The *database files* are located on disk drives that are shared between the 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:** An instance does not include database files. This means you can start up an instance without mounting the database files.

Each instance has a unique:

- SGA
- background processes
- ORACLE_SID registry entry

Different instances on different nodes can have the same system ID (SID). Oracle Corporation recommends each node in the domain having a unique SID to identify its instance. For example, the first node, called the *primary node*, uses the SID OPS1; the second node uses the SID OPS2 to identify its instance; and so on.

All nodes have the same components. The primary node is simply just the first node in cluster.

- set of redo logs

All instances share:

- the same set of data files
- the same control file

An instance contains:

- at least the following background processes:
  - PMON
  - SMON
  - DBWR
  - LGWR
- additional background processes:
  - LCK*n* to manage 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 that require exceptionally high throughput of instance lock requests. The single lock process per instance, LCK0, is usually sufficient for most systems.
  - LMON to monitor the health of the IDLM.
  - LMD0 to process lock requests
- the Integrated Distributed Lock Manager (IDLM) component in the SGA, an area for global locks and resources. This area was formerly part of the external Distributed Lock Manager (DLM).

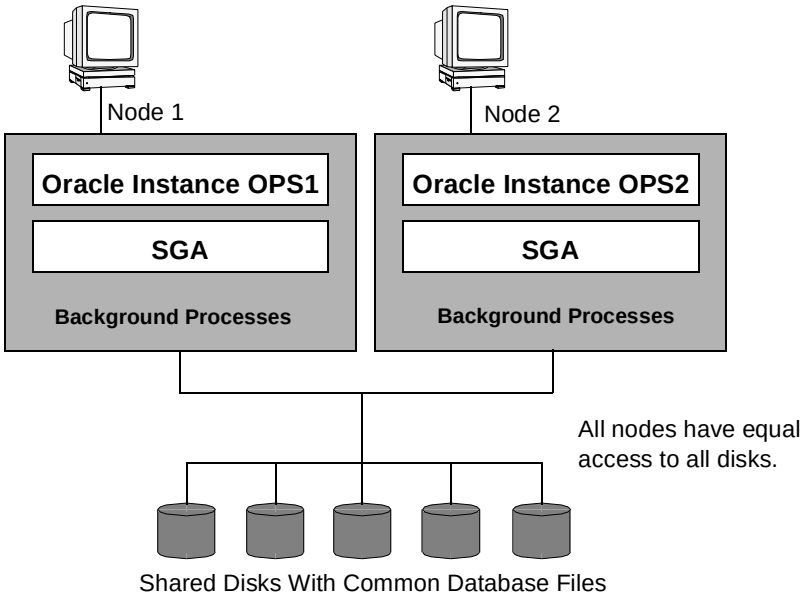
Additional Information: For more information on Oracle8 database processes and memory structures, see *Oracle8 Server Concepts*.

Database Files

A database is logically divided into tablespaces that contain all data stored in the database. Tablespaces, in turn, are made up of one or more data files. With Oracle Parallel Server, all participating instances access the same database files.

Figure 1–3, "Oracle Parallel Server Instance Architecture" shows the relationship between two Oracle instances and the shared disks on which the database files are stored:

Figure 1–3 Oracle Parallel Server Instance Architecture



Oracle Parallel Server Features

Oracle Parallel Server has the following features:

Feature	Description
Integrated Distributed Lock Manager (IDLM)	Maintains a list of system resources and provides locking mechanisms to control allocation and modification of Oracle resources. Every process interested in the database resource protected by the IDLM must open a lock on the resource. **Additional Information:** See the "Integrated Distributed Lock Manager" on page 1-11.
Parallel Cache Management (PCM)	Provides instance locks (with minimal use of the IDLM) that cover one or more data blocks of any class: data block, index blocks, undo blocks, segment headers, and so on. Oracle Parallel Server uses these instance locks to coordinate access to shared resources. The IDLM maintains the status of the instance locks **Additional Information:** See the "Parallel Cache Management" on page 1-12.
Oracle Parallel Query (OPQ)	Ships queries between nodes so that multiple nodes can execute on behalf of a single query. **Additional Information:** See the "Oracle Parallel Query" on page 1-15.

Integrated Distributed Lock Manager

The Integrated Distributed Lock Manager (IDLM) maintains a list of system resources and provides locking mechanisms to control allocation and modification of Oracle *resources*. Resources are structures of data. The IDLM does not control access to tables or anything in the database itself. Every process interested in the database resource protected by the IDLM must open a lock on the resource.

Oracle Parallel Server uses the IDLM facility to coordinate concurrent access to resources, such as data blocks and rollback segments, across multiple instances. The Integrated Distributed Lock Manager facility has replaced the external Distributed Lock Manager which was used in earlier releases of Oracle Server.

The IDLM uses the LMON and LMD*n* processes. LMON manages instance and processes deaths and associated recovery for the IDLM. In particular, LMON handles the part of recovery associated with global locks. The LMD*n* process handles remote lock requests (those which originate from another instance).

The IDLM:

- Provides a distributed architecture. In the distributed architecture each node in the cluster (or each instance of an Oracle database) participates in global lock management and manages a piece of the global lock database. The lock database is distributed among all the participants.
- Provides continual service and maintains the integrity of the lock database in the event of multiple node and instance failures.
- Maintains information about the locks on all nodes that are interested in a given resource. In this situation, the IDLM usually nominates one node to manage all relevant information about the resource and its locks.
- Performs deadlock detection.
- Synchronizes modifications made to the Oracle database files so that no changes are lost.
- Synchronizes disk block access between multiple nodes.

The IDLM is a resource manager and, thus, does *not* control access to the database.

Example A node in a cluster needs to modify block number *n* in the database file. At the same time, another node needs to update the same block *n* to complete a transaction.

Without the IDLM, *both* nodes update the same block at the same time. With the IDLM, only one node is allowed to update the block. The other node must wait. The IDLM ensures that only one instance has the right to update a block at any one time. This provides data integrity by ensuring that all changes made are saved in a consistent manner.

Interaction with PGMS

The IDLM uses PGMS to determine which instances are active. When the instance is started, the LMON and LMD*n* processes are started and the IDLM registers with PGMS. The IDLM deregisters with PGMS when the database is shutdown.

Parallel Cache Management

Parallel Cache Management (PCM) provides instance locks (with minimal use of the IDLM) that cover one or more data blocks of any class: data block, index blocks, undo blocks, segment headers, and so on. Oracle Parallel Server uses these instance locks to coordinate access to shared resources. The IDLM maintains the status of the instance locks.

PCM locks ensure cache coherency by forcing instances to acquire a lock before modifying or reading any database block. PCM locks allow only one instance at a time to modify a block. If a block is modified by an instance, the block must first be written to disk before another instance can acquire the PCM lock, read the block, and modify it.

Example If node 1 needs access to data that is currently in node 2's buffer cache, node 1 can submit a request to the IDLM. Node 2 then writes the needed blocks to disk. Only then is Node 1 notified by the IDLM to read updated and consistent data from the disk.

PCM Lock Implementation

You use the initialization parameter GC_FILES_TO_LOCKS to specify the number of PCM locks which cover the data blocks in a data file or set of data files. The smallest granularity is one PCM lock per data block; this is the default. PCM locks usually account for the greatest proportion of instance locks in a parallel server.

PCM locks are implemented in two ways:

PCM Lock	Description
hashed locks	Hashed locks are pre-allocated and statically assigned to one or more blocks at startup time. Hashed locks offer the advantage of no overhead for lock allocation when a data block is accessed. The first instance which starts up creates an IDLM resource and an IDLM lock (in null mode) on the IDLM resource for each hashed PCM lock. The first instance initializes each lock. The instance then proceeds to convert IDLM locks to other modes as required. When a second instance requires a particular IDLM lock, it waits until the lock is available and then converts the lock to the mode required. The total number of locks that can be allocated is limited by system resources. This usually means that multiple blocks have to be covered by the same lock. In other words, there is a low lock granularity. This might result in false pinging. The startup of the instance also requires more time, since all the lock resources have to be allocated at startup time. Typically, hashed locks are never released; each will stay in the mode in which it was last requested. If the lock is required by another. You can, however, specify releasable hashed locks by using the R option with the GC_FILES_TO_LOCKS parameter. Releasable hashed PCM locks are taken from the pool of GC_RELEASABLE_LOCKS.

PCM Lock	Description
fine-grain locks	This is the default implementation. With fine-grain locking, locks are dynamically allocated at block-access time. The resources for the lock are only allocated during the time the lock is needed and are released when the lock is released. This makes it possible to achieve very high-lock granularity. If resource minimization is the goal, fine-grain locks can also cover multiple blocks, but are still allocated dynamically. Since locks are allocated only as required, the instance can start up much faster than with hashed locks. An IDLM resource is created and an IDLM lock is obtained only when a user actually requests a block. Once a fine-grain lock has been created, it can be converted to various modes as required by various instances. Typically, fine-grain locks are releasable: An instance can give up all references to the resource name during normal operation. You can, however, allocate fixed locks in a fine grained manner with the GC_ FILES_TO_LOCKS parameter. Creating a 1 to 1 ratio of locks to blocks creates DBA locking.

Additional Information: See Chapter 9, Parallel Cache Management Instance Locks" and Chapter 15, "Allocating PCM Instance Locks", of the *Oracle8 Parallel Server Concepts and Administration* guide.

It is possible to have both fine-grain locking and hashed locking enabled at the same time.

Below is a comparison of both PCM locks:

Hash PCM Locks	Fine-Grain PCM Locks
■ Allocated at instance startup, resulting in a slower startup. ■ Released only at instance shutdown. ■ Statically hashed to blocks at startup time, requiring more memory.	■ Allocated when user requests a block, resulting in faster instance startup. ■ Dynamically re-used by other blocks, requiring less memory.

Choosing a Lock

Use the table below to choose a PCM lock:

When to use hashed locks...	When to use fine-grain locks...
■ Data is mostly read-only.	■ Small amount of data is updated by many instances.
■ Data can be partitioned.	■ There is not enough memory for the configuration of hash locking.
■ Large set of data is modified by a relatively small set of instances.	

Oracle Parallel Query

With the Oracle Parallel Query (OPQ), Oracle can divide the work of processing certain types of SQL statements among multiple query server processes.

Architectural Overview

When parallel execution is not being used, a single server thread performs all necessary processing for the sequential execution of a SQL statement. For example, to perform a full table scan (such as SELECT * FROM EMP), one thread performs the entire operation.

OPQ performs the operations in parallel using multiple parallel processes. One process, known as the *parallel coordinator,* dispatches the execution of a statement to several *parallel server processes* and coordinates the results from all the server processes to send the results back to the user.

The parallel coordinator breaks down execution functions into parallel pieces and then integrates the partial results produced by the parallel server processes. The number of parallel server processes assigned to a single operation is the degree of parallelism for an operation. Multiple operations within the same SQL statement all have the same degree of parallelism

OPQ on Oracle Parallel Server

Oracle Parallel Server provides the framework for the Parallel Query Option to work between nodes. The OPQ behaves the same way in Oracle with or without the Parallel Server Option. The only difference is that Oracle Parallel Server enables OPQ to ship queries between nodes so that multiple nodes can execute on behalf of a single query. Here, the server breaks the query up it into smaller operations that

run against a common database which resides on shared disks. Because it is performed by the server, this parallelism can occur at a low level of server operation, rather than at an external SQL level.

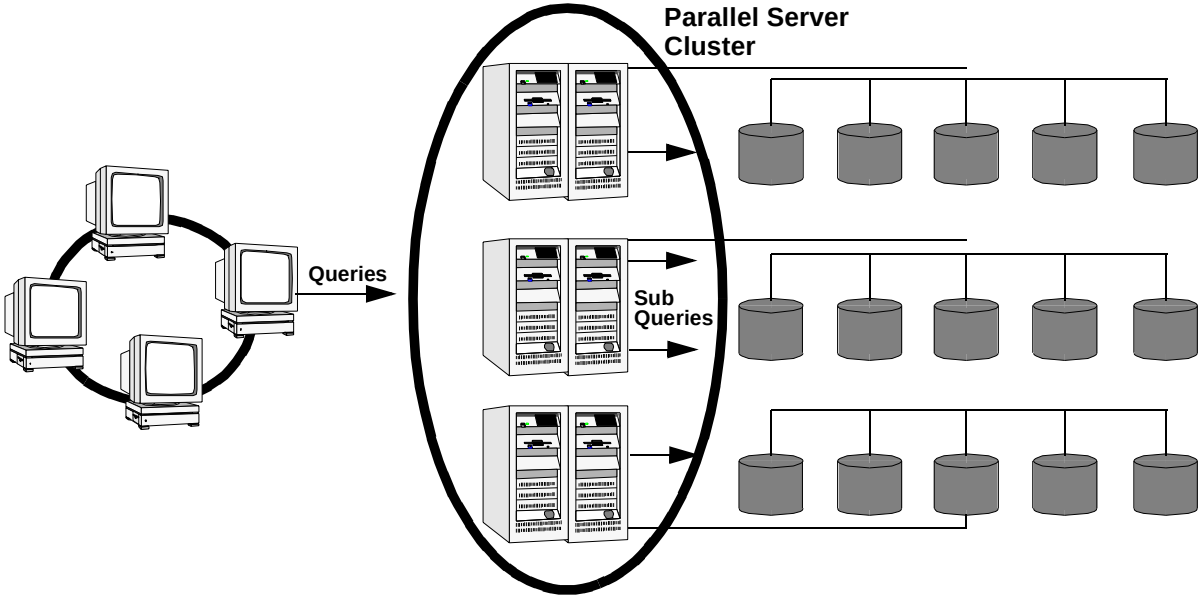
In some applications, an individual query often consumes a great deal of CPU resource and disk I/O (unlike most online insert or update transactions). To take advantage of multi-processing systems, the data server must parallelize individual queries into units of work which can be processed simultaneously.

If the query were not processed in parallel, disks would be read serially with a single I/O. A single CPU would have to scan all rows in a table. With the query parallelized, disks are read in parallel, with multiple I/Os.

Several CPUs can each scan a part of the table in parallel, and aggregate the results. Parallel query benefits not only from multiple CPUs but also from greater I/O bandwidth availability.

OPQ can run with or without the Oracle Parallel Server. Without the Oracle Parallel Server option, OPQ cannot perform multi-node parallelism. Oracle Parallel Server optimizes Oracle8 Enterprise Edition running on clustered hardware, using a parallel cache architecture to avoid shared memory bottlenecks in OLTP and decision support applications.

Figure 1–4 Parallel Query Operation



OPQ Statement

OPQ within Oracle Parallel Server performs parallelism within a node and among nodes via the parallel query slave processes on each node.

A sample SQL statement is shown below:

query broken up between *n* nodes

query broken up *n* times for each node

```
SELECT /*+ PARALLEL <OBJ$, 2, 2> */ * FROM OBJ$;
```

parallel query statement

After you have run a query, you can use the information derived from V$PQ_SYSSTAT to view the number of slave processes used, and other information for the system.

Additional Information: See Chapter 22, "Parallel Execution", of the *Oracle8 Server Concepts* guide.

2

System Requirements

This chapter describes the system requirements of Oracle Parallel Server. Specific topics discussed are:

- Hardware Requirements
- Software Requirements
- Shared Disk Subsystem

Hardware Requirements

Each Oracle8 Parallel Server Node

Each node in a cluster requires the following hardware:

- Personal computer (PC) with Pentium / Pentium Pro processor that has passed certification
- Windows NT-compatible network interface card (NIC). The NIC must be supported by Microsoft
- external shared hard disk
- 72 MB RAM minimum; 96 MB RAM recommended

Oracle Enterprise Manager Repository Database

Dedicated PC with Windows NT (fulfilling the above node requirements) as the repository database with a minimum of 32 MB RAM or 64 MB RAM if the repository database is also running Oracle Enterprise Manager.

Oracle Enterprise Manager Console

Additionally, a Windows NT, Windows 95 or Windows 98 console (or client machine) where Oracle Enterprise Manager resides requires the following hardware:

- Intel 486 PC or better
- Windows NT or 95 compatible network adapter
- VGA video (SVGA strongly recommended)
- 32 MB RAM recommended for Oracle Enterprise Manager
- 32 MB RAM for the Oracle Enterprise Manager Diagnostics Pack (optional)

Software Requirements

Each Oracle8 Parallel Server Node

Each node in a cluster requires the following software:

- ❏ Windows NT 4.0 Server
- ❏ Microsoft TCP/IP
- ❏ Oracle8 Enterprise Edition
- ❏ Net8 Server and Net8 Client
- ❏ Oracle8 Parallel Server Option
- ❏ Oracle8 Parallel Server Manager Server
- ❏ Operating System Dependent (OSD) layer from a vendor that has passed certification
- ❏ 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`

Oracle Enterprise Manager Repository Database

Oracle Enterprise Manager repository database machine requires:

- ❏ Oracle8 Enterprise Edition
- ❏ Net8 Server and Net8 Client

Oracle Enterprise Console

The Oracle Enterprise Manager console workstation requires:

- ❏ Net8 Client
- ❏ Oracle Enterprise Manager
- ❏ Diagnostics Pack for Oracle Performance Manager

  This feature is optional but recommended for most deployments.

Shared Disk Subsystem

Oracle Parallel Server requires a shared disk subsystem to contain shared partitions that are raw. All Oracle8 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 and control file, but data files are shared by each instance in the cluster. However, log files must be accessible / readable by other instances.

3

Performing Pre-Installation Tasks

This chapter describes required pre-installation tasks. Specific topics discussed are:

Raw Partition Overview

Because no true distributed file system support exists on Windows NT, the mechanism Oracle uses to allow nodes running different instances to access shared disks is to bypass the NT file system by using raw partitions. The shared disks are then a collection of unformatted raw devices.

Raw Partition

A raw partition is a portion of a physical disk that is accessed at the lowest possible level. I/O to a raw partition offers approximately a 5% to 10% performance improvement over I/O to a partition with a file system on it.

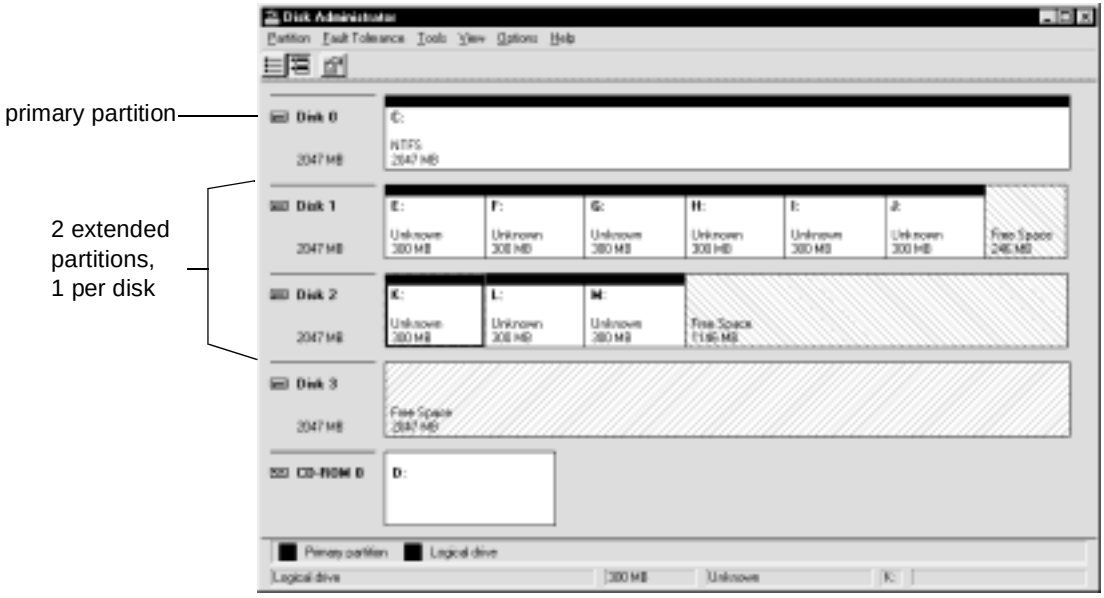
A raw partition is created when an extended partition is created and logical partitions are assigned to it without applying any formatting. The Windows NT Disk Administrator application allows you to create an extended partition on a physical drive.

Extended Partition

An extended partition points to raw space on the disk that can be assigned multiple logical partitions for the database files. An extended partition avoids the four-partition limit by allowing you to define large numbers of logical partitions to accommodate applications using Oracle8 Enterprise Edition. Logical partitions can then be given symbolic link names to free up drive letters. We create an extended partition in this chapter and link the drive letters to symbolic link names in Chapter 5, "Configuring Oracle Parallel Server".

The *Disk Administrator* window in Figure 3–1, "Disk Administrator" shows four disks, two of the disks having an extended partition.

Figure 3–1 Disk Administrator



This disk....	Contains...
Disk 0	a primary partition
Disk 1	an extended partition with six logical partitions and 246 MB of free space
Disk 2	an extended partition with three logical partitions and 1146 MB of free space
Disk 3	an unformatted partition

Note: You can tell whether a partition is formatted or unformatted by the direction of the diagonal lines. A formatted partition's lines display from top left to the bottom (\ \), and an uninitialized partition's lines display from top right to the bottom (/ /).

Disk Definition

Windows NT defines each disk drive found at startup with the following naming convention:

\Device\Harddisk*m*\Partition*n*

where Harddisk*m* is the number of the physical drive, and Partition*n* is a logical partition number, as shown in the *Disk Administrator* window (in Figure 3–1, "Disk Administrator"). Harddisk*m* starts at 0, and Partition*n* starts at 1. Partition0 has a special meaning in that it has access to the whole disk. For example, the first logical partition (E:) on the second physical drive in Figure 3–1, "Disk Administrator" has the following entry:

\Device\Harddisk1\Partition1

The first logical partition on a system (normally the C drive) has the following entry:

\Device\Harddisk0\Partition1

Raw Partition Definition

Raw partitions are of two types:

- Physical Disk
- Logical Partition

Note: Although you can use physical disks, Oracle Corporation recommends using logical partitions and creating symbolic links (as described in section "Step 2: Assign Symbolic Links to Each Logical Partition" on page 5-2).

Physical Disk

A physical disk represents the entire disk. It points to \Device\Harddisk*x*\Partition0. Windows NT automatically creates a symbolic link name of \\.\PhysicalDrive*x*, where *x* is the number corresponding to your hard disk drive number in the *Disk Administrator*. The *x* matches the *x* in \Device\Harddisk*x*\Partition0.

\\.\PhysicalDrive*x* is automatically defined by Windows NT for every hard disk in the machine. For example, a machine with three hard disks will have the following names defined by Windows NT:

\\.\PhysicalDrive0

\\.\PhysicalDrive1

\\.\PhysicalDrive2

Internally, these names expand to:

\\.\PhysicalDrive0 = \Device\Harddisk0\Partition0

\\.\PhysicalDrive1 =\Device\Harddisk1\Partition0

\\.\PhysicalDrive2 =\Device\Harddisk2\Partition0

Partition0 is special in that it represents the entire physical disk regardless of any partitioning scheme on that disk. On all disks recognized by Windows NT, the Disk Administrator writes a signature on the first block of all disks. To avoid overwriting that block, Oracle skips the first block of a physical raw partition which is used for an Oracle data file.

Logical Partition

A logical partition is a partition created by the *Disk Administrator* that points to a drive other than \Device\Harddisk*x*\Partition0. Logical partitions are initially assigned names with drive letters (\\.*drive_letter*:) and typically re-assigned symbolic link names (\\.*symbolic link name*). For example, \\.\D: may be assigned a symbolic link name of \\.\OPS_SYS01. Regardless of whether a drive letter or symbolic link name is used, logical partitions are defined to represent a specific partition in a disk rather than the entire disk. Internally, these names may expand to:

\\.\D:= \Device\Harddisk2\Partition1

\\.\OPS_SYS01= \Device\Harddisk3\Partition2

Drive letters can be assigned to specific partitions using the *Disk Administrator*, while symbolic link names can be assigned using a utility such as DOSDEV.EXE, which is available with the Windows NT Resource Kit, or the SETLINKS utility (shipped with the Parallel Server Option).

Note: Oracle does *not* skip the first block of a logical raw partition used for an Oracle data file.

Physical Disk and Logical Partition Considerations

Please consider the following when deciding which raw partition to use:

- Physical disks are automatically defined by Windows NT to represent the entire disk, and should *never* be defined by the user.
- Logical partitions need to be defined by the user to represent a specific partition in a disk. These partitions should be logical partitions or drives contained in an extended partition. They should *never* be defined as Partition0.
- Using an entire disk (Partition 0) for an Oracle data file and using a partition which occupies the entire disk for an Oracle data file is *not* the same thing. Even when a partition occupies the entire disk, there is still a small space on the disk that is not part of the partition.
- If you are using an entire disk for an Oracle data file (Partition 0), use the pre-defined physical raw names which Windows NT provides.
- Use a logical partition if you are using a specific partition and it occupies the entire disk.
- If using a specific partition created with the *Disk Administrator*, define and use a symbolic link name rather than a logical partition number (even if it occupies the entire disk).

Frequently Asked Questions

Question: What is the impact if I have created logical partitions, but defined physical disk convention names for them. For example:

\\.\PhysicalDriveSYS01 = \Device\Harddisk2\Partition1

\\.\PhysicalDriveUSR01 =\Device\Harddisk3\Partition1

Answer: Oracle database will handle the data file using the physical disk convention, even though it really is a logical partition. This will not cause any data corruption or loss, as long as you continue using the physical disk naming conventions. Oracle Corporation recommends you convert to the logical partition at your earliest convenience. See "Compatibility Issues" on page 3-7.

Question: What is the impact if I have created logical names representing Partition0. For example:

\\.\OPS_SYS01 = \Device\Harddisk1\Partition0

Answer: This will pose severe problems, as the Disk Administrator typically writes a signature into the first block of every disk, and consequently may overwrite a portion of the data file header.

This may also cause data loss. Never use Partition0 with the logical partition convention. See "Compatibility Issues" on page 3-7 for information on rebuilding your Oracle database with the proper conventions.

Question: How do I transfer the contents of any raw partition to a standard file system for backup purposes?

Answer: Use the Oracle provided OCOPY to copy data to/from a raw partition for both physical and logical raw conventions. See "Compatibility Issues" on page 3-7 for further information.

Compatibility Issues

The physical and logical partition conventions are not compatible with one another due to the extra block that is skipped for physical raw conventions. This also means you cannot simply do an OCOPY from a physical disk to a logical partition, as the contents of these partitions are incompatible.

To convert from a physical convention to a logical convention, you must:

1. Perform a full database export to a (local) file system.
2. Create logical partitions and define logical names for these partitions.
3. Re-create the database using the new logical partitions.
4. Perform the full database import to the newly created database.

If your database installation uses physical disk conventions with logical partitions, Oracle Corporation recommends converting to the logical partition conventions at your earliest convenience using the steps above.

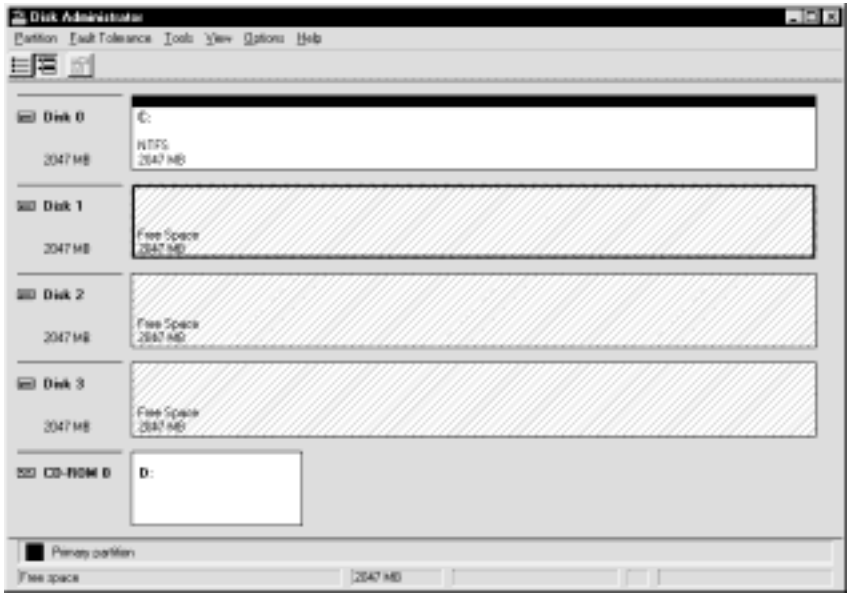
Step 1: Create an Extended Partition

Only one extended partition can be created per disk. You can use the free space in the extended partition to create multiple logical partitions or use all or part of it when creating volume sets or other kinds of volumes for fault-tolerance purposes.

To create an extended partition:

1. Shut down all nodes except the *primary* node.
2. On the *primary* node, choose Start > Programs > Administrative Tools > Disk Administrator.

   The *Disk Administrator* window appears:



   Note the lines display diagonally from top right to the bottom left, indicating unpartitioned devices.

3. Select an area of free space in an extended partition on a disk that is on the shared disk subsystem by clicking the mouse.

   Oracle recommends you use the entire disk.

4. Choose Create Extended... from the Partition menu.

The Disk Administrator displays the minimum and maximum sizes for the extended partition:

5. Use the default maximum size, then click OK.

> **Note:** Changes you have made are not saved until you choose Commit Changes Now Commit Changes Now from the Partition menu or quit the *Disk Administrator*.

The extended partition is created.

Note the lines now display diagonally from top left to bottom right, indicating the partition is an extended partition.

Step 2: Create Logical Partitions In an Extended Partition

After an extended drive is created, you must assign logical partitions to it. Logical partitions are assigned letters of the alphabet.

To create logical partitions in an extended partition:

1. On the *primary* node, select an area of free space in an extended partition by clicking the mouse on it.
2. Choose Create from the Partition menu.

   The *Disk Administrator* window displays the minimum and maximum sizes for the logical partition:

3. Enter the size of the logical partition, then click OK.

   The size is dependent on how large you want your log files and data files to be. Add 2MB to this size for overhead. If you plan on using the default database OPS.SQL creation script (to be used later) located in *ORACLE_HOME*\OPS to create your database later, 300 MB is more than enough.

   Additional Information: See "Step 7: Create an Oracle Parallel Server Database" on page 5-15 for more information about OPS.SQL.

4. Repeat Steps 1-3 until you have nine logical partitions.

   Attention: A two-node cluster requires 9 drives, and each additional node requires 2 additional drives to accommodate log files.

5. Choose Commit Changes Now from the Partition menu.

   A confirmation dialog appears, informing you changes have been made to the disk.

6. Click Yes to acknowledge the message.

   A dialog box appears, informing you the disks have been updated successfully.

7. Click OK.

8. Choose Close from the Partition menu.

9. Reboot all of the other nodes.

10. Choose Start > Programs > Administrative Tools > Disk Administrator on *all* the nodes.

On all nodes except the primary node, a dialog box appears, informing you the disk configuration has been detected. If you do not see this dialog box, it does not mean the disk configuration has not changed.

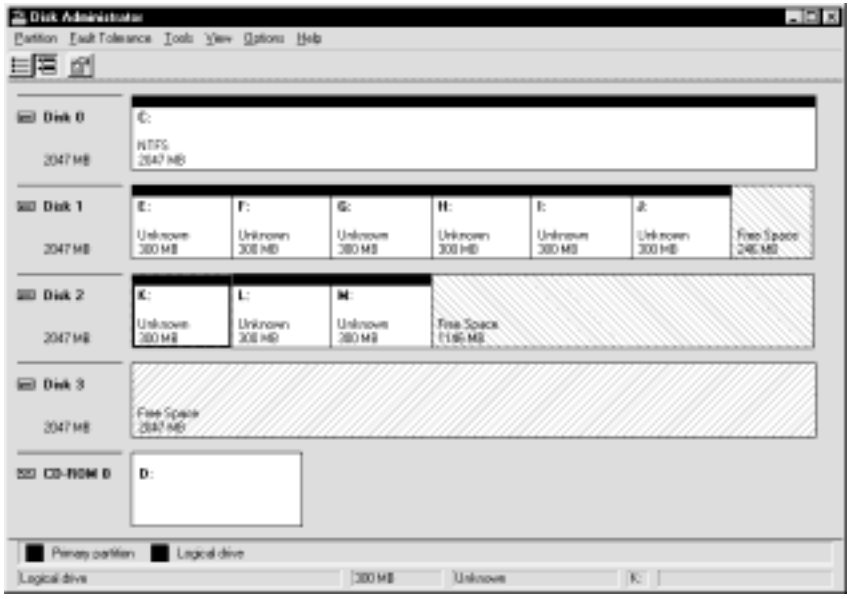
11. Click OK.

The extended partition and the logical partitions are now seen by all of the nodes.

Note: The hard disks that access the shared area may be different from node to node. The logical partitions, however, that reside on the hard disks must be identical on all nodes.

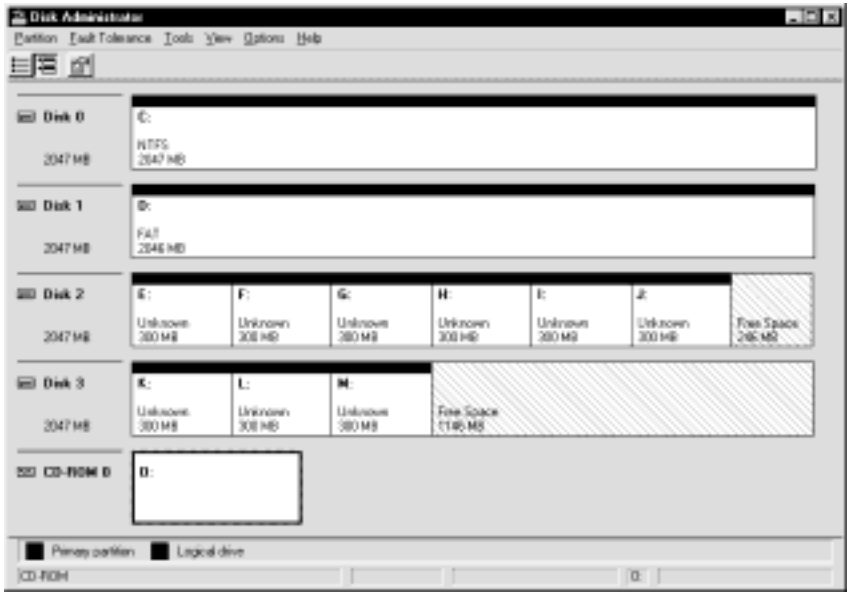
For example, Node 1 may look like:

Figure 3–2 Node 1 Logical Partitions



Node 2 may look like:

Figure 3–3 Node 2 Logical Partitions



On Node 1, the logical partition E is defined as \Device\Harddisk1\Partition1. On Node 2, it is defined as \Device\Harddisk2\Partition1. Even though these map to different disks, they map to the same space on the shared disk.

Step 3: Unassign Drive Letters

To free up other drive letters for other purposes, such as mapping network drives, it is important to also unassign the drive letters from the logical partitions on each partition. Oracle Parallel Server does not need to access a partition through a drive letter.

Note: A two-node cluster requires 9 logical partitions, for example, with drive letters E through M. Each additional node requires two additional drives to accommodate the log files.

Note: If your disks have been previously configured, the logical partitions may already be unassigned. If this the case, you can skip to Step 6 of this procedure.

To unassign drive letters:

1. On *each* node, select a logical partition.
2. Choose Assign Drive Letter from the Tools menu

   The *Assign Drive Letter* dialog box appears:

3. Choose *Do not assign a driver letter* and click OK.

   A confirmation dialog box appears.
4. Click Yes to confirm change.
5. Repeat Steps 1-4 for each logical partition on *each* node.

6. For each node in the cluster, write down the hard disk number(s) and the number of the partition (starting at 1) for that drive.

> **Note:** The hard disks that access the shared area may be different from node to node. The logical partitions, however, that reside on the hard disks must be identical from node to node.

Oracle Corporation recommends using a worksheet similar to the following.

	Node 1	Node 2	Node 3	Node 4
Hard Disk*x*	Partitions *x-x*	Partitions *x-x*	Partitions *x-x*	Partitions *x-x*
Hard Disk*x*	Partitions *x-x*	Partitions *x-x*	Partitions *x-x*	Partitions *x-x*

For example, the worksheet for the *Disk Administrator* windows shown in Figure 3–2, "Node 1 Logical Partitions" and Figure 3–3, "Node 2 Logical Partitions" of this chapter would look like:

	Node 1	Node 2
Hard Disk1	Partitions 1-6	n / a
Hard Disk2	Partitions 1-3	Partitions 1-6
Hard Disk3	n / a	Partitions 1-3

7. Choose Close from the Partition menu on all nodes.

   The *Disk Administrator* application exits.

8. Reboot each node.

9. Continue to Chapter 4, "Installing Oracle Parallel Server".

4

Installing Oracle Parallel Server

This chapter describes the installation and configuration tasks to create an operating Oracle Parallel Server. Specific topics discussed are:

- Oracle Parallel Server and Multiple Oracle Homes
- Step 1: Install Oracle8 Enterprise Edition and Oracle Parallel Server on Nodes
- Step 2: Install OSD Files

Oracle Parallel Server and Multiple Oracle Homes

When working with multiple Oracle homes with Oracle Parallel Server, keep the following in mind:

OraclePGMSService	OraclePGMSService supports multiple Oracle homes, but is only called from the last installed Oracle home. When the first Oracle home is installed with the Oracle Parallel Server Option, the OraclePGMSService service is created and the PGMS executable is installed in *ORACLE_HOME*\BIN\PGMS.EXE. When a second Oracle home is installed with the Oracle Parallel Server Option, PGMS.EXE is installed in the second Oracle home, and the OraclePGMSService only calls the PGMS.EXE from the second Oracle home (not the first).
Operating System Dependent (OSD) layer	Only one copy of the OSD layer can exist.

Additional Information: See Chapter 6, "Using Multiple Oracle Homes", of the *Oracle8 Enterprise Edition Getting Started for Windows NT*.

Step 1: Install Oracle8 Enterprise Edition and Oracle Parallel Server on Nodes

To install Oracle8 Enterprise Edition and Oracle Parallel Server:

1. Install Oracle8 Enterprise Edition on each of the nodes in the cluster, following instructions in *Oracle8 Enterprise Edition Installation for Windows NT* CD-ROM insert, and perform these tasks:

   a. Choose Oracle Parallel Server Option from the *Select Catridges and Options* dialog box.

   Note: The location of the Oracle Home can be different on each of the nodes.

   b. Choose None from the *Select a Starter Database Configuration* dialog box:

2. If you plan to install Oracle Enterprise Manager, ensure the Oracle Intelligent Agent service OracleAgent is installed on the nodes. The Oracle Intelligent Agent allows Oracle Enterprise Manager to communicate with the database.

Below is a a description of the Oracle Parallel Server components installed when the Oracle Parallel Option is chosen:

This component...	Installs...
Oracle Parallel Server Option	■ Oracle Parallel Server files ■ DLM files ■ SQL scripts and Oracle files for creating a four instance database ■ OraclePGMSService service ■ Oracle Parallel Server Manager Server (see below)
Oracle Parallel Server Manager Server	OPSM files needed to manage instances from individual nodes. **Additional Information:** See "Managing Instances Using OPSM" on page 7-8.

Step 2: Install OSD Files

A vendor-supplied Operating System Dependent (OSD) layer that passed certification must be installed after Oracle Parallel Server Option is installed.

To create the Oracle Parallel Server instances and the OSD registry entries:

1. For all nodes, see your OSD vendor documentation for instructions on installing OSD files.
2. Continue to Chapter 5, "Configuring Oracle Parallel Server" to configure Oracle Parallel Server.

5

Configuring Oracle Parallel Server

This chapter describes the configuration tasks to create an operating Oracle Parallel Server. Specific topics discussed are:

Step 1: Copy Initialization Files

The initialization files, INIT_COM.ORA and INIT*SID*.ORA are installed in the *ORACLE_HOME*\OPS directory. To use Oracle Parallel Server Manager and to avoid specifying the PFILE parameter when starting the database (STARTUP PFILE=C: ORANT\OPS\INIT*SID*.ORA), copy these file to the *ORACLE_HOME*\DATABASE file where initialization files are normally located.

To change the location of the initialization files:

1. Copy INIT_COM.ORA and INIT*SID*.ORA form the *ORACLE_HOME*\OPS directory to *ORACLE_HOME*\DATABASE directory.
2. Modify the INIT*SID*.ORA file so the IFILE points to the new *ORACLE_HOME*\DATABASE directory.

Step 2: Assign Symbolic Links to Each Logical Partition

The Oracle SETLINKS utility creates symbolic links to raw logical partitions.

For example, with the SETLINKS utility, you can create a symbolic link called OPS_log1t1 to the second logical partition on the second physical disk:

```
OPS_log1t1 \Device\Harddisk2\Partition2
```

The SETLINKS utility comes with the following ASCII files, with symbolic link names already associated with raw partitions. Each node requires two log files. If you have more than a three to four cluster, additional symbolic link names must be linked to the files:

This File...	Contains...
ORALINK1.TBL	symbolic link names already associated with raw partitions for a *two*-node cluster
ORALINK2.TBL	symbolic link names to accommodate a third node's additional log files
ORALINK3.TBL	symbolic link names to accommodate a fourth node's additional log files

Figure 5–1 ORALINK1.TBL

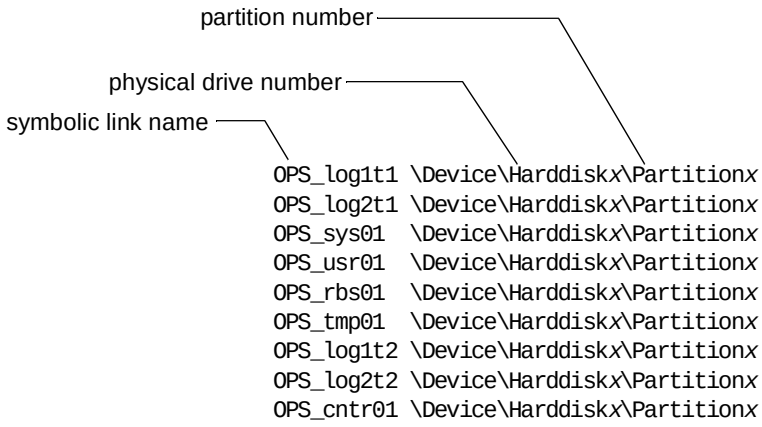


Figure 5–2 ORALINK2.TBL

```
OPS_log1t3 \Device\Harddiskx\Partitionx
OPS_log2t3 \Device\Harddiskx\Partitionx
```

Figure 5–3 ORALINK3.TBL

```
OPS_log1t4 \Device\Harddiskx\Partitionx
OPS_log2t4 \Device\Harddiskx\Partitionx
```

The following is a description of the symbolic link names:

Symbolic link name...	File Description	Symbolic link name used in the file *ORACLE_HOME*...
OPS_log1t1	recovery log file	\OPS\OPS.SQL
OPS_log2t1	recovery log file	\OPS\OPS.SQL
OPS_sys01	system data file	\OPS\OPS.SQL
OPS_usr01	user data file	\OPS\OPS.SQL
OPS_rbs01	rollback data file	\OPS\OPS.SQL

Symbolic link name...	File Description	Symbolic link name used in the file *ORACLE_HOME*...
OPS_tmp01	temporary data file	\OPS\OPS.SQL
OPS_log1t2	first log file for thread 1	\OPS\OPS.SQL
OPS_log2t2	second log file for thread 2	\OPS\OPS.SQL
OPS_cntr01	control file	\DATABASE\ INIT*SID*.ORA, INIT_ COM.ORA

To create symbolic links to the logical partitions:

1. On the *primary* node, edit the partition and hard disk numbers appropriately in the ORALINK1.TBL file located in *ORACLE_HOME*\OPS to match the disk numbers and partitions number you wrote down in "Step 3: Unassign Drive Letters" on page 3-13. If the cluster has a third and/or fourth node, edit ORALINK2.TBL (for the third node) and ORALLINK3.TBL for the fourth node.

   You can use a worksheet similar the one below to assist with the process:

Symbolic Link	Node 1	Node 2
OPS_log1t1	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*
OPS_log2t1	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*
OPS_sys01	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*
OPS_usr01	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*
OPS_rbs01	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*
OPS_tmp01	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*
OPS_log1t2	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*
OPS_log2t2	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*
OPS_cntr01	Harddisk*x* Partition*x*	Harddisk*x* Partition*x*

   For example, the worksheet for the two nodes shown in section "Step 2: Create Logical Partitions In an Extended Partition" on page 3-9 looks likethe following:

Symbolic Link	Node 1	Node 2
OPS_log1t1	Harddisk1 Partition1	Harddisk2 Partition1
OPS_log2t1	Harddisk1 Partition2	Harddisk2 Partition2
OPS_sys01	Harddisk1 Partition3	Harddisk2 Partition3
OPS_usr01	Harddisk1 Partition4	Harddisk2 Partition4
OPS_rbs01	Harddisk1 Partition5	Harddisk2 Partition5
OPS_tmp01	Harddisk1 Partition6	Harddisk2 Partition6
OPS_log1t2	Harddisk2 Partition1	Harddisk3 Partition1
OPS_log2t2	Harddisk2 Partition2	Harddisk3 Partition2
OPS_cntr01	Harddisk2 Partition3	Harddisk3 Partition3

Additional Information: See Chapter 3, "Performing Pre-Installation Tasks", for further information about the *Disk Administrator*.

WARNING: Do not change the symbolic link names in OPSLINK1.TBL. If you do, symbolic link names must be modified for the following files:

- ***ORACLE_HOME*\OPS\OPS.SQL**
- ***ORACLE_HOME*\DATABASE\INIT_COM.ORA**

2. Run the ASCII input file, ORALINK1.TBL, through the SETLINKS utility as shown below:

```
C:\> CD ORACLE_HOME\OPS
C:\ORACLE_HOME\OPS> SETLINKS /F:ORALINK1.TBL
```

SETLINKS maps the drives to the symbolic label names:

```
Oracle Corporation. Copyright (c) 1997. All rights reserved.
Created Link:
Created Link:OPS_log1t1 = Device:\Device\Harddisk1\Partition1
Created Link:OPS_log2t1 = Device:\Device\Harddisk1\Partition2
```

```
Created Link:OPS_sys01 = Device:\Device\Harddisk1\Partition3
Created Link:OPS_usr01 = Device:\Device\Harddisk1\Partition4
Created Link:OPS_rbs01 = Device:\Device\Harddisk1\Partition5
Created Link:OPS_tmp01 = Device:\Device\Harddisk1\Partition6
Created Link:OPS_log1t2 = Device:\Device\Harddisk2\Partition1
Created Link:OPS_log2t2 = Device:\Device\Harddisk2\Partition2
Created Link:OPS_cntr01 = Device:\Device\Harddisk3\Partition3
Dos devices updated successfully.
```

If the cluster has third or fourth node, SETLINKS must be run again with OPSLINK2.TBL (for the third node) or OPSLINK3.TBL for the fourth node.

```
C:\> CD ORACLE_HOME\OPS
C:\ORACLE_HOME\OPS> SETLINKS /F:ORALINK2.TBL
C:\ORACLE_HOME\OPS> SETLINKS /F:ORALINK3.TBL
```

3. Make sure the drives have been mapped with the correct names as shown below:

```
C:\ORACLE_HOME\OPS> SETLINKS /D
```

SETLINKS shows how the symbolic link names have been mapped:

```
Oracle Corporation. Copyright (c) 1997. All rights reserved.
OPS_log1t1  = \Device\Harddisk1\Partition1
OPS_log2t1  = \Device\Harddisk1\Partition2
OPS_sys01   = \Device\Harddisk1\Partition2
OPS_usr01   = \Device\Harddisk1\Partition2
OPS_rbs01   = \Device\Harddisk1\Partition5
OPS_tmp01   = \Device\Harddisk1\Partition6
OPS_log1t2  = \Device\Harddisk2\Partition1
OPS_log2t2  = \Device\Harddisk2\Partition2
OPS_cntr01  = \Device\Harddisk2\Partition3
...
```

4. Repeat Steps 1-3 on the other nodes.

Removing or Ignoring Links

Links can be removed or ignored by changing the contents of the ORALINK*x*.TBL files and running it through SETLINKS again.

Ignoring Links

Insert a '#" at the beginning of a line to denote a comment, which is ignored by SETLINKS.

Removing an Existing Symbolic Llink

List a symbolic link without a device path in the ORALINK*x*.TBL file.

For example:

```
OPS_log1t1
OPS_log2t1 \Device\Harddisk1\Partition2
OPS_sys01  \Device\Harddisk1\Partition3
OPS_usr01  \Device\Harddisk1\Partition4
OPS_rbs01  \Device\Harddisk1\Partition5
OPS_tmp01  \Device\Harddisk1\Partition6
OPS_log1t2 \Device\Harddisk2\Partition1
OPS_log2t2 \Device\Harddisk2\Partition2
OPS_cntr01 \Device\Harddisk2\Partition3
```

This removes the OPS_log1t1 link and creates the other links specified.

Step 3: Configure OSD Layer

Note: When following the steps below, note that some tasks are performed on the *primary node only*, while other tasks are performed on *each node*.

To configure the Operating System Dependent (OSD) components:

See your OSD vendor documentation for instructions about configuring the OSD components, making sure the:

- domain name you specify is identical to the database name defined by the DB_NAME parameter in the INIT_COM.ORA file located in *ORACLE_HOME*\DATABASE. The database name is OPS by default.
- instance number you specify must be one less than the digit defined by the INSTANCE_NUMBER parameter in the INIT*SID*.ORA file located in *ORACLE_HOME*\DATABASE.
- machine names you specify are alpha-numeric

CAUTION: When configuring domains, two instances cannot reside on the same machine and be in the same domain.

Step 4: Create Services

After the OSD components have been configured, an OracleService*SID* for each instance must be created.

To create OracleService*SID*:

1. On *each* node, go to the *ORACLE_HOME*\BIN directory:

   ```
   C:\> CD ORACLE_HOME\BIN
   ```

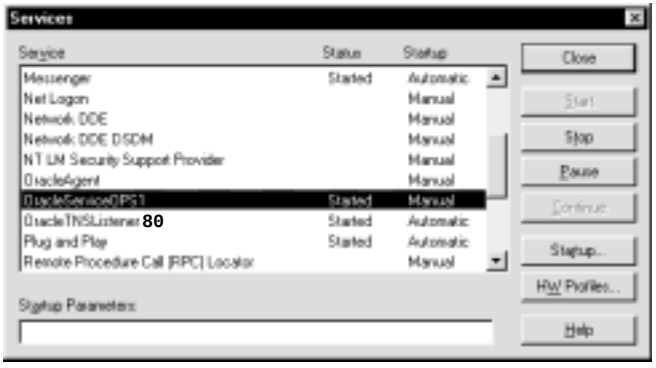
2. On *each* node, use the CRTSRV batch file to create a unique service corresponding to the instance of the node. This batch file also sets up the service so that each time it started, the OraclePGMSService is started automatically.

   ```
   C:\ORACLE_HOME\BIN> CRTSRV SID
   ```

   For example, to create OracleServiceOPS1 on the primary node:

   ```
   C:\ORACLE_HOME\BIN> CRTSRV OPS1
   ```

3. Verify OracleService*SID* exists by choosing Services from the Control Panel. The *Services* window appears:



4. Some vendors may require changing the password for OracleService*SID* and Oracle*HOME_NAME*TNSListener80 to the Administrator's password for the machine:

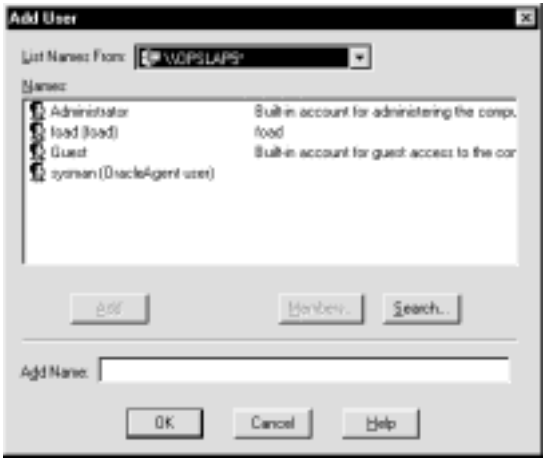
Additional Information: See Chapter 6, "Using Multiple Oracle Homes", in the *Oracle8 Enterprise Edition Getting Started for Windows NT* for a description of *HOME_NAME* and multiple Oracle homes.

---

a. Double-click the service. The *Service* dialog box appears:

b. Select the This Account option button.

c. Select the "..." button. The *Add User* dialog box appears:



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

e. Click OK. The *Service* dialog box reappears.

f. Enter the NT administrator's password in the Password and Confirm Password fields.

g. Click OK.

The *Services* window reappears.

h. Click Close to exit the Services window.

5. Stop OracleService*SID*:

- From the MS-DOS command line, enter:

```
C:\> NET STOP OracleService SID
```

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

Step 5: Add the ORACLE_SID Entry to the Registry

The SID for an instance must be set in the registry to allow instance recognition when you access a database from a node. When you connect to a database, Oracle Parallel Server will be able to resolve the SID name.

> **Note:** If the SID is not set in the registry, Oracle Parallel Server tries to see if the SID is set as an environment variable. If it cannot find it, the SID name defaults to ORCL. ORCL is not a valid SID for an Oracle Parallel Server. Therefore, you cannot connect with it.

Oracle Parallel Server requires the ORACLE_SID registry value to be manually added to the following location:

If you have...	Parameter is located in...
One Oracle home directory	HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE
Two or more Oracle home directories	HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*ID* where *ID* is incremented for each additional Oracle home directory on your machine.

To add the ORACLE_SID registry value:

1. From *each* node, start the registry in either one of two ways:

- From the command line, enter:

```
C:\> REGEDT32
```

- Choose Start > Run, enter REGEDT32 in the Open field, then click OK.

The *Registry Editor* window appears:

2. Double-click the SOFTWARE key.
3. Navigate to the proper Oracle home key.
4. Choose Add Value from Edit menu. The *Add Value* dialog box appears:

5. In the Value Name field, enter ORACLE_SID.
6. In the Data Type box, select REG_SZ.
7. Click OK. The *String Editor* dialog box appears:

8. Enter the SID of the instance for the node you are on, then click OK.

   Registry Editor adds ORACLE_SID.
9. Choose Exit from the Registry menu.

   The registry exits.

Step 6: Configure the Network

Configure the nodes and clients for connections. Net8 clients communicate with Oracle8 servers through service names, which are easy to remember aliases for database addresses.

Each node must have the following configuration files in *ORACLE_HOME*\NET80\ADMIN:

Configuration File	Description	Configuration Requirements

LISTENER.ORA	Includes addresses of all network listeners on a server, the system identifiers (SIDs) of the databases for which they listen, and various control parameters used by the network listener.	On each node, LISTENER.ORA must contain an entry for the database name and SID of the node in the SID_LIST section. Do not make an entry for GLOBAL_DBNAME. For example:
		`SID_LIST_LISTENER =`
		`(SID_LIST =`
		`(SID_DESC =`
		`(SID_NAME = OPS)`
		`)`
		`(SID_DESC =`
		`(SID_NAME = ops1)`
		`)`

TNSNAMES.ORA	Includes a list of service names of databases.	A service name for the database and the name of the node's instance.

The OPSCONF utility *must* be used to generates a TNSNAMES.ORA file with the service names already configured for all the nodes in a cluster and a LISTENER.ORA file with SID_NAME parameter already set.

The steps to perform to configure the network include:

Step 6a: Stop the Network Listener

Step 6b: Configure the Service Names

Step 6c: Start the Network Listener

Step 6d: Test the Configuration

Step 6a: Stop the Network Listener

The network listener must be stopped on each node.

To stop the network listener:

- From the MS-DOS command line, enter:

  ```
  C:\> NET STOP OracleHOME_NAMETNSListener80
  C:\> NET STOP OracleHOME_NAMETNSListener80LSNR
  ```
  (where *LSNR* is the non-default network listener name)

- From the Control Panel's *Services* window, select Oracle*HOME_NAME*TNSListener80 (the service name if you are using the default network listener name LISTENER), or Oracle*HOME_NAME*TNSListener80*LSNR* (where *LSNR* is the non-default network listener name), then click Stop.

  A message appears, confirming you want to stop the network listener. Click Yes.

Step 6b: Configure the Service Names

Configuration files can be created with the OPSCONF utility. The OPSCONF utility creates properly configured files for all nodes, including:

- TNSNAMES.ORA
- LISTENER_*NODE_HOSTNAME*.ORA (created for each node in the cluster)
- TOPOLOGY.OPS defines the parallel servers and instances in the network. This file is not needed if Oracle Enterprise Manager's auto-discovery feature is working.

These files are created on the node in *ORACLE_HOME*\OPS. After creation, you need to move the files to their proper configuration file names and to the proper machines.

To create the configuration files:

1. On one of the nodes, go to the *ORACLE_HOME*\BIN directory:

   ```
   C:\> CD ORACLE_HOME\BIN
   ```

2. Run OPSCONF:

   ```
   C:\ORACLE_HOME\BIN> OPSCONF
   ```

   OPSCONF runs in interactive mode, prompting you with configuration questions. Bold indicates user input:

   a. `Enter the name of the Oracle Parallel Server database.`

      database_name

      The database name is defined in the INIT_COM.ORA file in *ORACLE_HOME*\DATABASE. The database name is defaulted to OPS.

   b. `Enter the port number of the listener (Default=1526).`

      port_number

      Press **[Enter]** to accept the default value.

   c. `Enter the port number of the Oracleagent (Default=1748).`

      Press **[Enter]** to accept the default value unless you have changed the port number.

   All the configuration files are created in *ORACLE_HOME*\OPS.

3. Move each LISTENER_*node_hostname*.ORA file to the *ORACLE_HOME*\NET80\ADMIN directory to the appropriate node, renaming it to LISTENER.ORA.

4. Copy TNSNAMES.ORA file to the *ORACLE_HOME*\NET80\ADMIN directory on each of the nodes, all client machines, and the Oracle Enterprise Manager console.

5. Move TOPOLOGY.OPS to the *ORACLE_HOME*\NET80\ADMIN directory on the Oracle Enterprise Manager console workstation, renaming it to TOPOLOGY.ORA. Or, merge it with your existing TOPOLOGY.ORA.

Step 6c: Start the Network Listener

The network listener must be started on each node.

To start the network listener:

- From the MS-DOS command line, enter:

  ```
  C:\> NET START OracleHOME_NAMETNSListener80
  C:\> NET START OracleHOME_NAMETNSListener80LSNR
  ```
  (where *LSNR* is the non-default network listener name)

- From the Control Panel's *Services* window, select Oracle*HOME_NAME*TNSListener80 (the service name if you are using the default network listener name LISTENER), or Oracle*HOME_NAME*TNSListener80*LSNR* (where *LSNR* is the non-default network listener name), then click Start.

Step 6d: Test the Configuration

To ensure the files are configured correctly:

1. On any node or client machine, start Server Manager. Enter:

   ```
   C:\> SVRMGR30
   ```

2. Connect to an instance:

   ```
   SVRMGR> CONNECT INTERNAL/PASSWORD@SERVICE_NAME
   ```

3. Server Manager returns a "*Connected*" message to the screen.

   If there is an error in connecting, you must troubleshoot your installation. Typically, there is a problem with the IP address, host name, or SID.

Step 7: Create an Oracle Parallel Server Database

Creating an Oracle Parallel Server database involves generating:

- database files and rollback segments
- tablespaces
- SYSTEM tables

CAUTION: The instructions below assume the domain you specified during the OSD configuration is identical to the database name defined in the INIT_COM.ORA file.

To create the Oracle Parallel Server database:

1. Start the OracleServiceOPS1 service

Note: Prior to starting a OracleService*SID* service, the OraclePGMSService service must be running. If you used the CRTSRV script in "Step 4: Create Services" on page 5-8, OraclePGMSService automatically starts when the OracleService*SID* service is started.

If you chose to create your services with another method, you can still have OraclePGMSService start up automatically with a OracleService*SID* service by entering the following at the command for each node:

```
C:\> OPSREG80 ADD SID
```

   - From the MS-DOS command line, enter:

     ```
     C:\> NET START OracleServiceOPS1
     ```

   - From the Control Panel's *Services* window, select OracleServiceOPS1, then click Start.

2. Connect to the instance and run the OPSALL.SQL script to create the database:

   ```
   C:\> CD ORACLE_HOME\OPS
   C:\ORACLE_HOME\OPS> SVRMGR30
   SVRMGR> @OPSALL.SQL
   ```

3. Enter the password for INTERNAL when prompted.

   The OPSALL.SQL script:

   - Calls the OPS.SQL script to create the Oracle Parallel Server database files, rollback segments, and tablespaces
   - Builds a system data dictionary
   - Creates users
   - Creates views
   - Creates redo log files for each instance

- Shuts down the database

Note: You may modify the OPS.SQL script in *ORACLE_HOME*\OPS to better suit your database needs.

---

4. Stop the OracleServiceOPS1 service:
   - From the MS-DOS command line, enter:

     ```
     C:\> NET STOP OracleServiceOPS1
     ```

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

Step 8: Start the Database in Parallel Mode

To start the Oracle Parallel Server database in parallel mode:

Note: When following the steps below, note that some tasks are performed on the *primary node only*, while other tasks are performed on *each node*.

---

1. On *each* node, delete the *ORACLE_HOME*\RDBMS80\TRACE*SID*LMON.TRC file.

   This file is re-created with new Oracle Parallel Server messages.

2. Start OracleService*SID* instance on *each* node.

Note: Prior to starting a OracleService*SID* service, the OraclePGMSService service must be running. If you used the CRTSRV script in "Step 4: Create Services" on page 5-8, OraclePGMSService automatically starts up when the OracleService*SID* service is started.

If you chose to create your services with another method, you can still have OraclePGMSService start up automatically with a OracleService*SID* service by entering the following at the command for each node:

```
C:\> OPSREG80 ADD SID
```

- From the MS-DOS command line, enter:

  ```
  C:\> NET START OracleService OPS1
  ```

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

3. On the *primary* node, start the database:

   ```
   C:\> SVRMGR30
   SVRMGR> CONNECT INTERNAL/PASSWORD
   SVRMGR> STARTUP
   ```

   The first instance to start up in shared mode determines the values of the global constant 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 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:

   ```
   C:\> SVRMGR30
   SVRMGR> CONNECT INTERNAL/PASSWORD
   SVRMGR> STARTUP
   ```

   Note: If PARALLEL_SERVER (the default) is set in the INIT_COM.ORA file, you do not have specify STARTUP PARALLEL.

   You also do not have specify the PFILE parameter when starting the database (STARTUP PFILE=C: ORANT\OPS\INIT*SID*.ORA) if you followed the procedure in "Step 1: Copy Initialization Files" on page 5-2.

5. On *each* node, ensure there is a "*Reconfiguration complete*" message in the *ORACLE_HOME*\RDBMS80\TRACE*SID*LMON.TRC file for the correct number of node, signifying the cluster was started without errors:

   Below is a sample of the trace file, indicating successful startup for two nodes:

   ```
   Thu Oct 30 15:22:59 1997
   Reconfiguration started
   ```

```
List of nodes: 1,2,
 Lock DB frozen
 Non-local Process blocks cleaned out
 Non-local Group blocks cleaned out
 Resources and locks cleaned out
 Traffic controller Initialized
 Communication channels reestablished
 Submitted all rdomain info
 Submitted all remote-lock requests
 Update rdomain variables
 Dwn-cvts replayed, VALBLKs dubious
 All grantable locks granted
Thu Oct 30 15:23:02 1997
Reconfiguration complete
```

Step 9: Verify InstancesAre Running

To verify instances are running:

1. On any node, enter the following:

```
C:\> SVRMGR30
SVRMGR> CONNECT INTERNAL/PASSWORD
SVRMGR> SELECT * FROM V$ACTIVE_INSTANCES;
```

The following output is returned:

```
INST_NUMBRE  INST_NAME
-----------  ---------
          1 node1:ops1
          2 node2:ops2
          3 node3:ops3
. .
```

> **CAUTION: If another node is reconfigured *after* this table has been accessed and this statement is issued again, only the last node whose database was started displays.**

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

```
C:\> SVRMGR30
SVRMGR> CONNECT SCOTT/TIGER
SVRMGR> 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.
```

3. On the one node, increase MILLER's salary by $1000 and commit the change:

```
SVRMGR> UPDATE EMP
SET SAL = SAL + 1000
WHERE ENAME = 'MILLER';
SVRMGR> COMMIT;
```

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

```
SVRMGR> SELECT * FROM EMP;
```

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

5. Continue to Chapter 6, "Installing and Configuring Oracle Parallel Server Manager", if you want to configure Oracle Parallel Server Manager with Oracle Enterprise Manager.

6

Installing and Configuring Oracle Parallel Server Manager

This chapter describes how to install and set up Oracle Parallel Server Manager (OPSM). Specific topics discussed are:

Overview of OPSM

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

In addition to managing parallel databases, OPSM allows 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.

This console requires the following components:

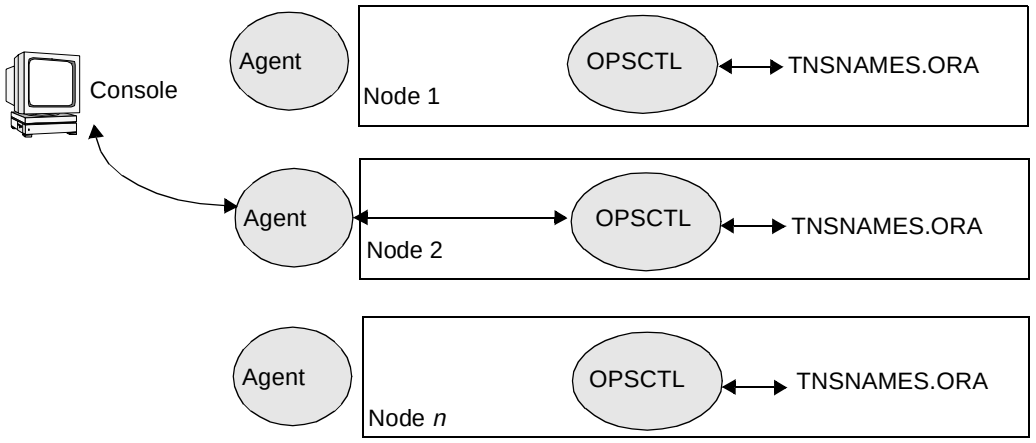
Component	Description
Oracle Enterprise Manager Console	Oracle Enterprise Manager Console works with the Oracle Intelligent Agent (installed on the nodes) to perform database administration from a single console. Oracle Enterprise Manager is installed on a Windows NT, Windows 95 or Windows 98 client.
(Optional) Oracle Performance Manager	The Oracle Enterprise Manager's Oracle Performance Manager allows you to choose from a variety of tabular and graphic performance statistics for parallel servers.
Repository Database	Oracle Enterprise Manager Console uses a repository database to store its information. This repository database is an Oracle8 database installed on the Oracle Enterprise Manager Console if it is a Windows NT machine, or on another Windows NT machine where a Windows NT, Windows 95 or Windows 98 Oracle Enterprise Manager Console can connect. The repository database *must* be on a separate machine from the nodes.

Component	Description
Oracle Intelligent Agent	The Oracle Intelligent Agent is installed on the nodes with the Oracle Enterprise Edition database. The Oracle Intelligent Agent is responsible for managing and completing tasks requested from Oracle Enterprise Manager Console. Once installed, the Oracle Intelligent Agent: ■ Listens for and responds to job requests initiated by the Oracle Enterprise Manager Console. ■ Schedules job requests (requests can include detecting and correcting abnormal conditions, performing standard database administration procedures, and monitoring events) executes the job request. **Note:** If you need to re-install the Oracle Intelligent Agent, make sure that the `ORACLE_HOME\NET80\AGENT\CONFIG\NMICONF.LST` file that is over-written contains an entry for `CONFOPS.TCL`. After auto-discovery is performed, the Oracle Intelligent Agent runs the scripts in the `NMICONF.LST` file. `CONFOPS.TCL` is required for OPSM.

OPSM uses the OPSCTL utility (installed on each node) to manage instances. This utility is run directly from the command line on a node, or called by the Oracle Enterprise Manager Console. OPSCTL gathers information about all the instances in a cluster from the TNSNAMES.ORA files. When OPSCTL is called by the Oracle Enterprise Manager Console, only one node's Intelligent Agent is used to communicate to OPSCTL. OPSCTL on that node then communicates to the other nodes through Net8.

Figure 6–1, "OPSCTL Architecture" shows how OPSCTL operates between the agent and the nodes.

Figure 6–1 OPSCTL Architecture



OPSM Requirements

The following requirements must be met in order to use OPSM from the Oracle Enterprise Manager Console:

The requirements for each node are:

Node Requirements	For additional information, see:
Ensure Oracle8 Enterprise Edition, Oracle Parallel Server Option, Oracle Parallel Server Manager Server, and Oracle Intelligent Agent are installed on each node in the cluster.	"Step 1: Install Oracle8 Enterprise Edition and Oracle Parallel Server on Nodes" in Chapter 4, "Installing Oracle Parallel Server"
The INIT_COM.ORA and INIT*SID*.ORA initialization files must located in *ORACLE_HOME*\DATABASE.	"Step 1: Copy Initialization Files" in Chapter 5, "Configuring Oracle Parallel Server"
Use OPSCONF to create a TNSNAMES.ORA file for each node. Each instance must have a service name specified for it, which is used in defining the database links for the instances.	"Step 6: Configure the Network" in Chapter 5, "Configuring Oracle Parallel Server"
The SIDs of the instances in the parallel server must be different.	Chapter 5, "Configuring Oracle Parallel Server"
In order to use the optional Oracle Performance Manager component, run OPS_8MON.SQL script.	"Step 3: Configure Oracle Performance Manager" on page 6-18

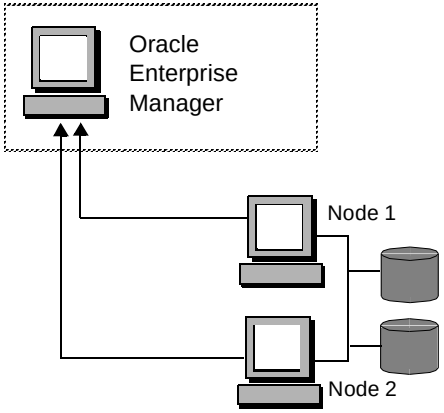
The requirements for the Oracle Enterprise Manager Console are:

Oracle Enterprise Manager Console Requirements	For additional information, see:
Install your Enterprise Manager version 1.6 software (with or without Oracle Diagnostics Pack option) on a Windows NT, Windows 95 or Windows 98 computer. Oracle Diagnostics Pack, which contains the Oracle Performance Manager application, is sold separately.	"Step 1: Install Oracle Enterprise Manager" on page 6-7.
If the Oracle Enterprise Manager Console is on separate machine than the repository database, ensure that the Net8 Client is installed and properly configured on the Console.	"Step 1: Install Oracle Enterprise Manager" on page 6-7. and "Step 2d: Create Configuration Files" on page 6-14.
Configure and connect to the repository database, including performing auto-discovery of your nodes.	"Step 2: Configure Oracle Enterprise Manager" on page 6-8.
To use the optional Oracle Performance Manager component, specify the user preferences for your Oracle Parallel Server nodes and your Oracle Parallel Server databases.	Chapter 2, “Oracle Parallel Server Management Operations,” of the *Oracle Parallel Server Management User’s Guide*.

Understanding Oracle Enterprise Manager Console Setup

Set up your Oracle Enterprise Manager Console environment in either of two ways.

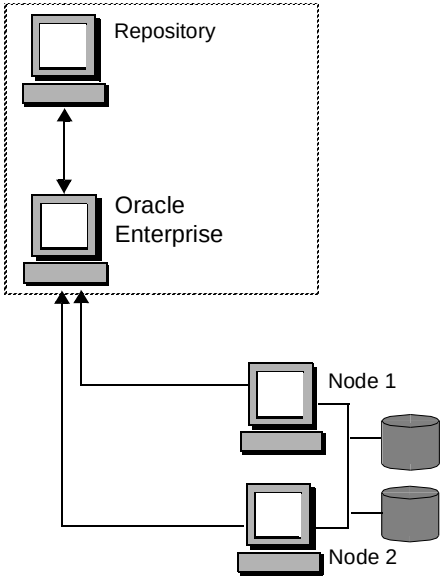
Environment 1: Console and Repository on same machine



The Console remotely manages the databases for both nodes. The Console is running on a Windows NT machine with an Oracle8 database installed that is only being used as a repository. The Oracle Intelligent Agent does *not* need to be running.

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

Environment 2: Console and repository on separate machines



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

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

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

Step 1: Install Oracle Enterprise Manager

You can install Oracle Enterprise Manager on a Windows NT, Windows 95 or Windows 98 machine.

Installing Oracle Enterprise Manager includes:

- Step 1a: Install Repository Database on Windows NT
- Step 1b: Install Oracle Enterprise Manager on Console Workstation.

Step 1a: Install Repository Database on Windows NT

The repository database is an Oracle8 Enterprise Edition database installed on the console if it is a Windows NT machine, or on another Windows NT machine where the Windows NT, Windows 95 or Windows 98 console can connect.

To install the repository database:

Install Oracle8 Enterprise Edition on a Windows NT Server by following the instructions in *Oracle8 Enterprise Edition for Windows NT Installation CD-ROM Insert* CD-ROM insert.

Step 1b: Install Oracle Enterprise Manager on Console Workstation

Oracle Enterprise Manager version 1.5 is installed on a console workstation, a Windows NT, Windows 95 or Windows 98 client which connects to the Oracle8 repository database. To install the Oracle Enterprise Manager:

1. Install Net8 Client (if you want Oracle Enterprise Manager Console to be on separate machine from the repository database) from the *Software Asset Manager* window after choosing the Custom Installation option.

   Additional Information: See the *Oracle8 Enterprise Edition for Windows NT Installation CD-ROM Insert* CD-ROM insert for complete installation instructions.

2. Install Oracle Enterprise Manager.

   Additional Information: See the *Oracle8 Enterprise Edition for Windows NT Installation CD-ROM Insert* CD-ROM insert for complete installation instructions.

3. Install the Oracle Diagnostics Pack to use the Oracle Performance Manager. Oracle Diagnostics Pack is on another CD from Oracle Enterprise Manager, requiring a separate license.

Additional Information: See the *Oracle Diagnostics Installation* CD-ROM insert for complete installation instructions.

Step 2: Configure Oracle Enterprise Manager

This table describes the tasks that must be performed to configure the two Oracle Enterprise Manager Console environments illustrated above. Each task is then described fully below:

Perform This Task...	On All Nodes?	On machine where Oracle Enterprise Manager Console and Repository machine both are?	On Oracle Enterprise Manager Console machine or Repository machine?
Step 2a: Start Oracle Services	Yes	No	No
Step 2b: Create a Repository User Account	No	Yes	Yes, perform this task on the Repository machine.
Step 2c: Create a Windows NT User Account	Yes	No	No
Step 2d: Create Configuration Files	No	No	Yes, perform this task on the Oracle Enterprise Manager Console machine to connect to the Repository machine.
Step 2e: Create the Repository and DiscoverServices	No	Yes	Yes, perform this task on the Repository machine.

Note: These instructions assume Oracle Enterprise Manager is already installed on a console workstation. See the installation CD-ROM insert to install Oracle Enterprise Manager and see "Step 1b: Install Oracle Enterprise Manager on Console Workstation" on page 6-7 to install OPSM Client.

Step 2a: Start Oracle Services

Start OracleService*SID*, Oracle*HOME_NAME*TNSListener80, OracleAgent and OraclePGMSService services are running on *each* node:

To start the Oracle services:

1. From the *Control Panel* window, double-click Services.

   The *Services* window appears.

2. Check the status of the services. A blank in Status column indicates that a service is not running.

   If all the services are already running, close the window and go to "Step 2b: Create a Repository User Account" on page 6-10. If a service is not running, to Step 3 in this procedure.

3. Start the service(s):

   Note: Prior to starting a OracleService*SID* service, the OraclePGMSService service must be running. If you used the CRTSRV script in "Step 4: Create Services" on page 5-8, OraclePGMSService automatically starts when the OracleService*SID* service is started.

   If you chose to create your services with another method, you can still have OraclePGMSService start up automatically with a OracleService*SID* service by entering the following at the command for each node:

   ```
   C:\> OPSREG80 ADD SID
   ```

   - From the MS-DOS command line, enter:

     ```
     C:\> NET START SERVICE
     ```

   - From the Control Panel's *Services* window, select a service, then choose Start.

   The SNMP_RW.ORA and SNMP_RO.ORA files located in ORANT\NET80\ADMIN are automatically generated when the Intelligent Agent is started. These files contain the listening address of the agent, the names of Net8 listener and Oracle database services it knows about, plus tracing parameters. Do not update these read-only files

Additonal Information: See Appendix A, "Configuration Files," of the *Oracle Enterprise Manager Configuration Guide*, for a description of SNMP_RO.ORA and SNMP_RW.ORA parameters.

Step 2b: Create a Repository User Account

You need to have a user account with DBA (database administrator) privileges set up on the Oracle8 repository database. Oracle Enterprise Manager places its repository in the default tablespace of the user account to which you are connected when creating the repository. Oracle Corporation recommends creating a new user account, creating a new tablespace, and the customizing the newly-created user to use the new tablespace as its default tablespace.

To create a repository user account on the Oracle8 repository database:

1. Start Server Manager and connect to the Oracle8 repository database:

   ```
   C:\> SVRMGR30
   SVRMGR> CONNECT SYSTEM/MANAGER
   ```

2. Create a special tablespace for the Oracle Enterprise Manager repository:

   ```
   SVRMGR> CREATE TABLESPACE REPOSIT_TABLESPACE DATAFILE 'PATH\DATAFILE' SIZE
   XM;
   ```

   where *X* is the tablespace size in megabytes; 20 MB is a good starting point.

3. Create a user account where the user's default tablespace is the newly-created tablespace:

   ```
   SVRMGR> CREATE USER USERNAME IDENTIFIED BY PASSWORD
   2> DEFAULT TABLESPACE REPOSIT_TABLESPACE;
   ```

4. Grant DBA privileges to this user account:

   ```
   SVRMGR> GRANT DBA TO USERNAME IDENTIFIED BY PASSWORD;
   ```

5. When done, enter:

   ```
   SVRMGR> EXIT
   ```

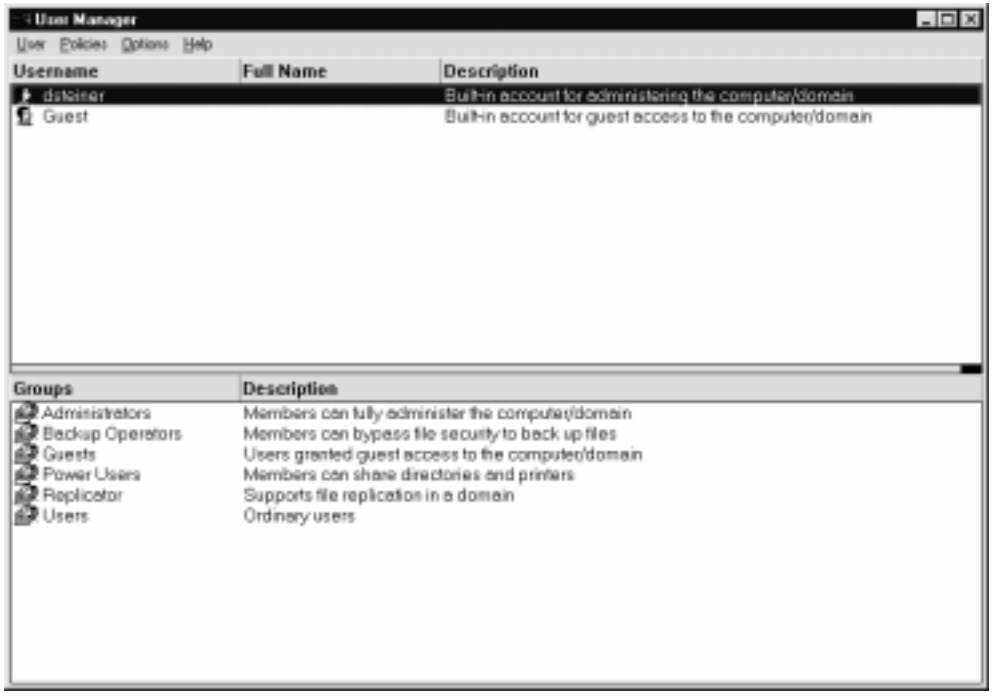
   Exiting Server Manager closes the database connection.

6. Follow Steps 1-5 for additional repository user accounts.

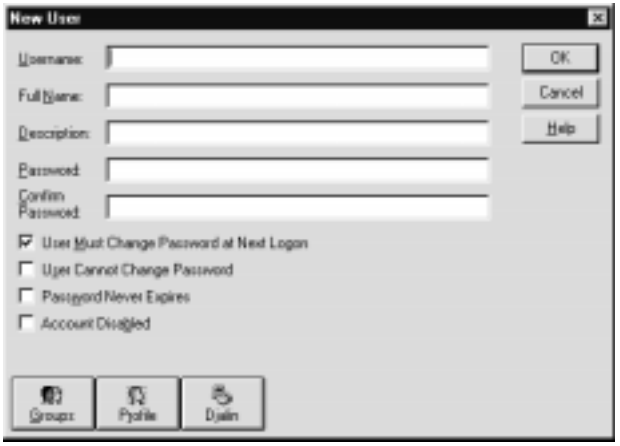
Step 2c: Create a Windows NT User Account

After a repository user account is created, an NT user account with the same user name as the repository user account must be created on the the individual Oracle Parallel Server nodes. 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:



2. Choose New User from the User menu. The *New User* dialog box appears:



3. Enter the appropriate information in the dialog box:

   a. Enter a user name that is the same as the one you entered in section "Step 2b: Create a Repository User Account" in the Username field.

   b. Enter the same password you entered in section Step 2b: Create a Repository User Account in both the Password and Confirm Password fields.

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

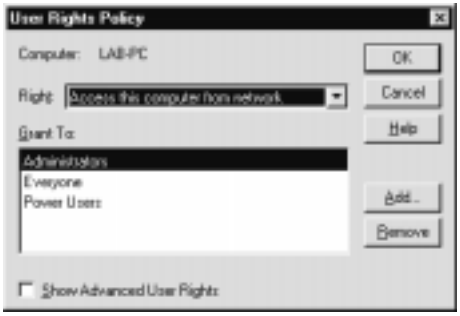
   d. Make the user a member of the Administrator's group by choosing Groups. The *Group Memberships* dialog box appears:

e. Select Administrators from the *Not member of* list box, then click <- Add. Administrators is added to the *Member of* list box.

f. 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 User Rights from the Policies menu. The *User Rights Policy* dialog box appears:



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

7. Select Users from the Grant To list box. If Users does not exist, create it:

a. Click Add. The *Add Users and Groups* dialog box appears:

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

c. Select Users from the Names list box, then click Add.

d. Click OK.

Users appears in the Grant To list box.

8. Click OK in the *User Rights Policy* dialog box.

The *User Manager* window re-appears.

9. Choose Exit from the User menu.

10. Repeat Steps 1-9 for any other repository user accounts you created.

Step 2d: Create Configuration Files

If Oracle Enterprise Manager is installed on a Windows NT, Windows 95 or Windows 98 client console separate from the machine on which the repository database is installed, you must configure the client machine so it can connect to the Oracle8 repository database. Use the Oracle Net8 Easy Config to configure your network.

Step 2e: Create the Repository and DiscoverServices

Before you can use Oracle Enterprise Manager Console, you need a repository on the console workstation where Oracle Enterprise Manager Console will run. The repository is a set of base tables in a database that contains status and environment information for your managed databases. A separate repository must be created for each user wanting to use Oracle Enterprise Manager Console.

The Oracle Enterprise Manager Console machine must also discover network services, such as databases, listeners, and nodes, to populate the Navigator tree. In addition to discovery, the Oracle Enterprise Manager Console must also be able to manage these services for the Job Scheduling and Event Management systems. This can be achieved with the Discovery Wizard.

When the agent starts up and is explicitly requested by the Oracle Enterprise Manager Console with the Navigator Discovery feature, the agent passes information from the Oracle Enterprise Manager Console to populate the Navigator tree.

To create a repository on the Oracle Enterprise Manager Console machine:

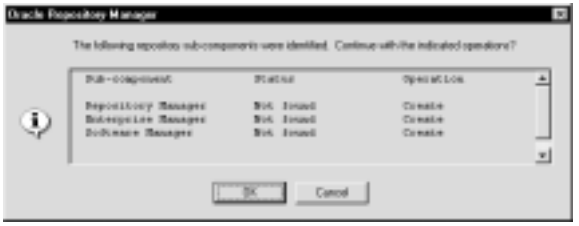
1. Choose Start > Programs > Oracle Enterprise Manager. The *Repository Login Information* dialog box appears:

2. Enter the following appropriate information, then click OK:

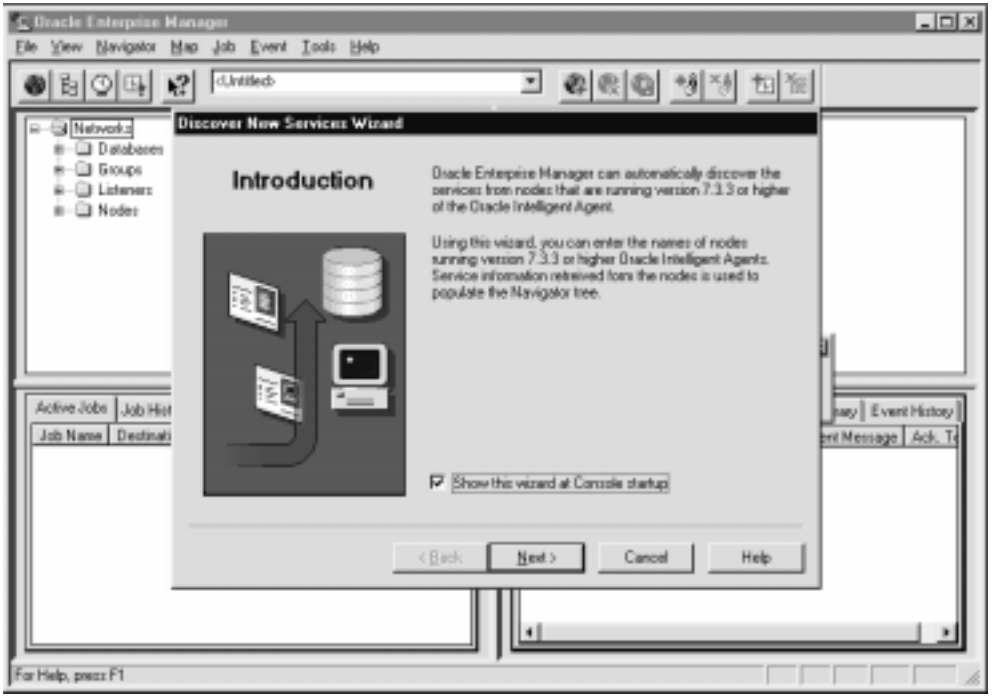
   a. Repository user name and password you created in "Step 2b: Create a Repository User Account" in Username

   b. Net8 service name (or database alias) of the repository database; only necessary to enter service name if you are connecting from a client console machine connecting to the Oracle8 repository database

The *Oracle Repository Manager* window appears, prompting you to build the repository:

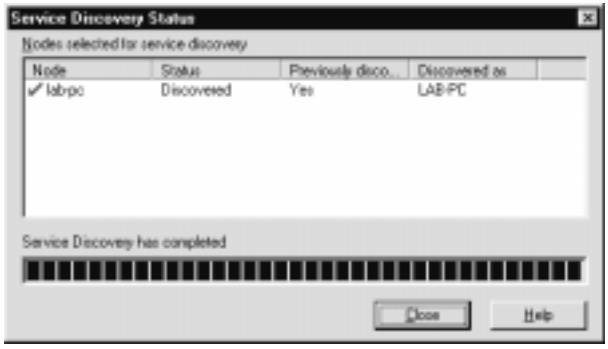


3. Click OK. The repository is built. A dialog box asking if you want a quick tour of Oracle Enterprise Manger appears.
4. Click Continue to dismiss this dialog box. The Discover New Services Wizard appears:

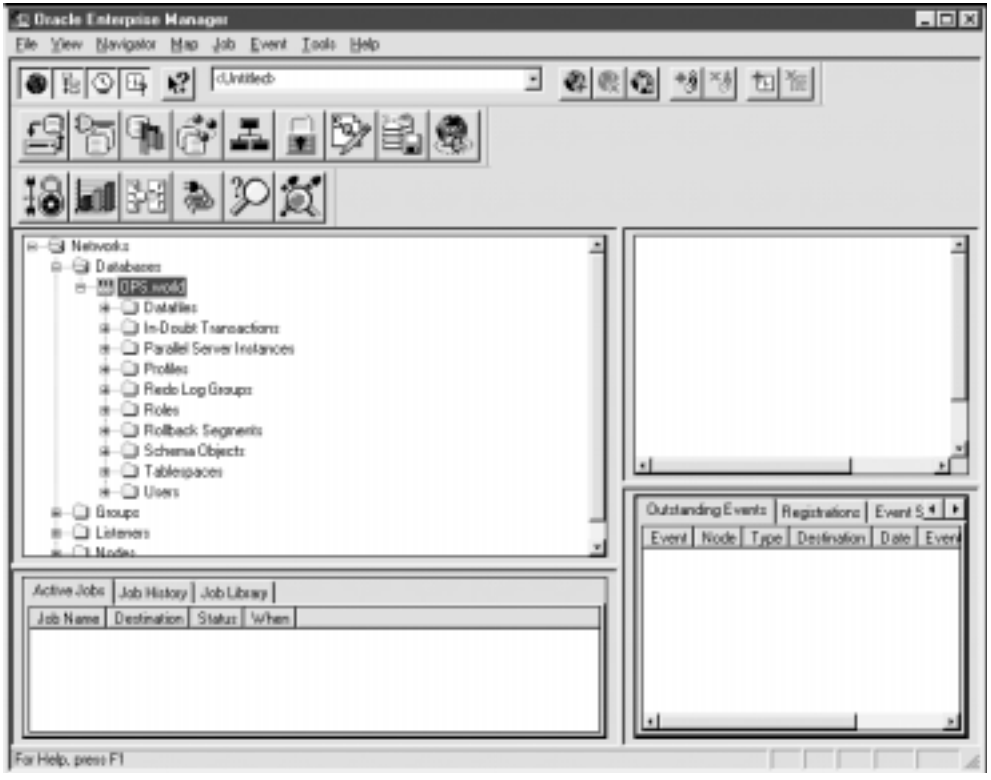


5. Use the wizard to discover services.

Services are discovered when you see the dialog box below with a status of *Discovered*:



6. Click Close. The *Oracle Enterprise Manager* Console appears with access to the Oracle Parallel Servers in the Navigator tree:



Additional Information: See the *Oracle Parallel Server Management User's Guide* for further information about managing Oracle Parallel Servers and parallel server instances from Oracle Enterprise Manager Console.

Step 3: Configure Oracle Performance Manager

The Oracle Enterprise Manager's Oracle Performance Manager allows you to choose from a variety of tabular and graphic performance statistics for parallel servers. The statistics represent the aggregate performance for all instances running on a parallel server. The statistics are displayed in individual charts and include information on data block pings, lock activity, file I/O, and session and user information.

Note: The Enterprise Manager Intelligent Agent is *not* required by the Oracle Performance Manager, as its functions are performed using a database connection to the parallel server.

To configure Oracle Performance Manager:

1. Enter the following at the MS-DOS command line from any node:

```
C:\> CD ORACLE_HOME\OPS
C:\ORACLE_HOME\OPS> SVRMGR30
SVRMGR> @OPS_8MON.SQL
```

2. Increase the OPEN_LINKS and DISTRIBUTED_TRANSACTAIONS parameters in the INIT_COM.ORA file by the number of instances in the parallel server:

Parameter	Description
OPEN_LINKS	Specifies the maximum number of concurrent open connections to remote databases in one session. The default is set to 4.
DISTRIBUTED_ TRANSACTIONS	Specifies the maximum number of distributed transactions in which this database can concurrently participate. The default is set to 16.

Note: By default these parameters are not stored in the INIT_COM.ORA file. To increase the value of these parameters, you must add them to the INIT_COM.ORA file.

Oracle Enterprise Manager Oracle Performance Manager can be invoked against any instance of the parallel server. Oracle Performance Manager will execute a PL/SQL procedure on this instance to query for performance data from all instances of the parallel server via database links, aggregating the results into a set of database tables, which are then displayed in graphical form by Oracle Performance Manager.

Accessing Oracle Performance Manager

To start Oracle Performance Manager:

Note: Oracle Performance Manager requires at least two instances in the Oracle Parallel Server be up and running.

---

1. Choose Start > Programs > Oracle Enterprise Manager > Enterprise Manager.

   The *Repository Login Information* dialog box appears:

2. Enter the following information, then click OK:
   - Repository user name
   - Password for the user name
   - Net8 service name of the repository database; only necessary to enter service name if you are connecting from a client console machine connecting to the Oracle8 repository database

3. Start the Oracle Performance Manager by:
   - Clicking on the Oracle Performance Manager application icon from the Diagnostics Pack launch palette.
   - Selecting Diagnostics Pack, and then Oracle Performance Manager from the Tools menu.

   The *Oracle Performance Manager* application appears.
4. Click on the Databases folder.
5. Double-click on a database instance.
6. Enter a database user name and password when prompted.

   The list of available charts appears.
7. Double-click Parallel Server.
8. Click any of the charts.

   The charts display in the right-hand window.

Additional Information: See the *Oracle Parallel Server Management User's Guide* for further information about managing Oracle Parallel Servers and parallel server instances from Oracle Enterprise Manager Console.

7

Administering Multiple Instances

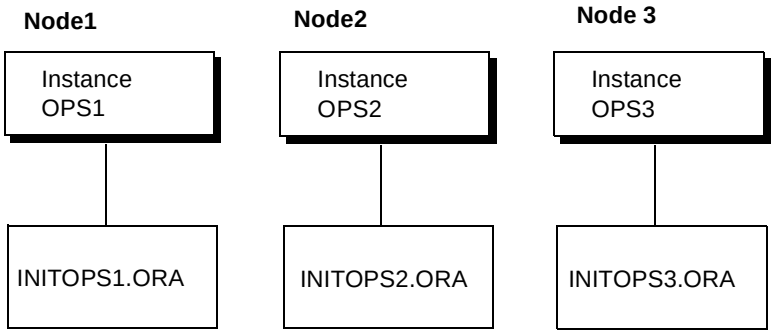
This chapter describes how instances are managed through initialization files and Oracle Parallel Server Manager (OPSM). Specific topics covered in this chapter are:

- Understanding the Initialization Files
- Setting Initialization Parameters for the Oracle Parallel Server
- Managing Instances Using OPSM

Understanding the Initialization Files

An initialization parameter file is an ASCII text file containing a list of parameters. Each node consists of an initialization parameter file named INIT*SID*.ORA with parameters unique for an instance and an initialization parameter file named INIT_COM.ORA with common parameters shared from node-to-node.

Figure 7–1 Instance Initialization Files



Purpose of INIT*SID*.ORA

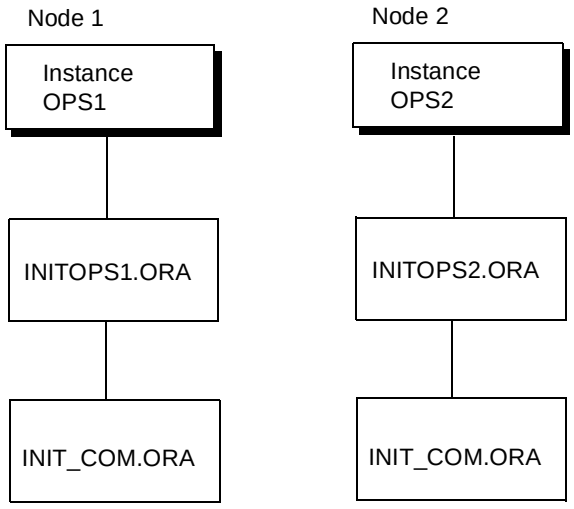
The INIT*SID*.ORA initialization parameter files point to INIT_COM.ORA file for common parameters and define the:

- unique instance number
- thread number for instance
- private rollback segments for an instance

Purpose of INIT_COM.ORA

INIT_COM.ORA is called by the individual parameter files through the IFILE parameter.

Figure 7–2 Common Initialization Files



In a parallel server, some initialization parameters must have the same values for every instance, whether individual or common parameter files are used. By referencing the common parameter file using the IFILE parameter within the individual parameter files, instances that have individual parameter files can use the correct parameter values for those that must be identical. This also allows individual parameter files with different values.

Figure 7–3, "INITOPS1.ORA" shows the contents of INITOPS1.ORA:

Figure 7–3 INITOPS1.ORA

```
instance_number=1
thread=1
rollback_segments=(RB1, RB2, RB3, RB4, RB5, RB6, RB7, RB8)
ifile=c:\orant\ops\init_com.ora
```

Figure 7–4, "INITOPS2.ORA" shows the contents of INITOPS2.ORA:

Figure 7–4 INITOPS2.ORA

```
instance_number=2
thread=2
rollback_segments=(RB9, RB10, RB11, RB12, RB13, RB14, RB15, RB16)
ifile=c:\ORANT\ops\init_com.ora
```

Note: During the installation process, these initialization files are automatically created.

Figure 7–5, "INIT_COM.ORA" shows a sample INIT_COM.ORA file:

Figure 7–5 INIT_COM.ORA

```
db_name=ops
db_files = 1024
control_files = '\\.\OPS_cntr01'
db_file_multiblock_read_count =  8
db_block_buffers =  200
shared_pool_size =  10000000
log_checkpoint_interval = 10000
processes =  59
parallel_max_servers = 5
log_buffer =  8192
sequence_cache_entries =  10
sequence_cache_hash_buckets =  10
max_dump_file_size = 10240
background_dump_dest=%RDBMS80%\trace
user_dump_dest=%RDBMS80%\trace
db_block_size = 2048
remote_login_passwordfile = shared
text_enable = TRUE
job_queue_processes = 2
job_queue_interval = 10
job_queue_keep_connections = false
distributed_lock_timeout = 300
distributed_transactions = 5
open_links = 4
parallel_server = true
```

Setting Initialization Parameters for the Oracle Parallel Server

This section describes the following:

- GC Global Cache Parameters
- Parameter Notes for Multiple Instances
- Identical Parameters on Each Instance

- Parameter Descriptions
- Location of Initialization Files
- Editing Initialization Files

GC Global Cache Parameters

Initialization parameters with the prefix GC (Global Cache) are relevant only for an Oracle Parallel Server. These parameters are specified in the INIT_COM.ORA file. Global cache parameter settings determine the size of the collection of global locks that protect the database buffers on all instances. The settings you choose affect the use of certain operating system resources.

Of the instances (OPS1, OPS2, and so on), 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 Server compares the values of the global cache 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 cache parameters.

The global cache parameters for an Oracle Parallel Server are:

Parameter	Description
GC_FILES_TO_LOCKS	Gives the mapping of hashed and fine-grain locks to blocks within each data file. The meaning of this parameter has changed. Previously, files not mentioned in this parameter (or files added later) were assigned the remaining hash locks. Files not mentioned in this parameter use DBA locking. You can now have multiple entries of GC_ FILES_TO_LOCKS.
GC_LCK_PROCS	Specifies the number of LCK background processes for one instance.
GC_RELEASABLE_LOCKS	Sets the number of locks which will be used for DBA locks.
GC_ROLLBACK_LOCKS	For each rollback segment, specifies the number of instance locks available for simultaneously modified rollback segment blocks.

Parameter Notes for Multiple Instances

Multiple instance issues concerning initialization parameters \are summarized in the following table:

Parameter	Parallel Server Notes
CHECKPOINT_PROCESS	In Oracle Parallel Server, your database can have more datafiles. To speed up checkpoints, enable the CHECKPOINT_PROCESS parameter.
DELAYED_LOGGING_BLOCK_ CLEANOUTS	If set to TRUE, this parameter can potentially reduce pinging between instances.
DML_LOCKS	This parameter must be identical on all instances only if set to zero.
INSTANCE_NUMBER	If specified, this parameter must have unique values for different instances.
LOG_ARCHIVE_FORMAT	You must include the thread number.
MAX_COMMIT_PROPAGATION_ DELAY	If you want commits to be seen immediately on remote instances, you may need to change the value of this parameter.
NLS_* parameters	This parameter can have different values for different instances.
PARALLEL_SERVER	To enable a database to be started in parallel server mode, this parameter must be set to TRUE in the initialization file.
PROCESSES	This parameter must have a value large enough to allow for all background processes and all user processes in an instance. Some operating systems can have additional DBWR processes. Defaults for the SESSIONS and TRANSACTIONS parameters are derived directly or indirectly from the value of the PROCESSES parameter. If you do not use the defaults, you may want to increase some of these parameter values to allow for optional background processes.
RECOVERY_PARALLELISM	To speed up the roll forward or cache recovery phase, you can set this parameter.
ROLLBACK_SEGMENTS	Specify the private rollback segments for each instance.
THREAD	If specified, this parameter must have unique values for different instances.

Identical Parameters on Each Instance

Certain initialization parameters are critical at database creation or affect certain database operations. These parameters must have the same value for every instance in an Oracle Parallel Server specified in each INIT_COM.ORA file for each instance. For example, the values of DB_BLOCK_SIZE and CONTROL_FILES must be identical for every instance. Other parameters can have different values for different instances; for example, INIT_SQL_FILES can have any value because it is ignored except when the database is created.

The following initialization parameters must have identical values for every instance in a parallel server:

CACHE_SIZE_THRESHOLD	LM_LOCKS (identical values recommended)
CONTROL_FILES	LM_PROCS (identical values recommended)
CPU_COUNT	LM_RESS (identical values recommended)
DB_BLOCK_SIZE	LOG_FILES
DB_FILES	MAX_COMMIT_PROPAGATION_DELAY
DB_NAME	PARALLEL_DEFAULT_MAX_INSTANCES
DML_LOCKS (must be identical only if set to zero)	PARALLEL_DEFAULT_MAX_SCANS
GC_FILES_TO_LOCKS	ROLLBACK_SEGMENTS
GC_LCK_PROCS	ROW_LOCKING
GC_ROLLBACK_LOCKS	SERIALIZABLE

Setting LM Parameters

Set values for the LM_* initialization parameters. Note that the resources, locks and processes configurations are per OPS instance. For ease of administration, these parameters should be consistent for all the instances.

Parameter	Description
LM_LOCKS	Number of locks. Where R is the number of resources, N is the total number of nodes, and L is the total number of locks, the following calculation is used: L = R + (R*(N - 1)) / N

Parameter	Description
LM_PROCS	Number of processes. The value of PROCESSES initialization parameter multiplied by the number of nodes.
LM_RESS	This parameter controls the number of resources that can be locked by the Lock Manager. This parameter covers the number of lock resources allocated for DML, DDL (data dictionary locks), and data dictionary cache locks + file and log management locks.

Parameter Descriptions

Chapter 18, "Administering Multiple Instances," of the *Oracle8 Parallel Server Concepts & Administration* includes descriptions of the initialization parameters:

Location of Initialization Files

The database for which the instance is started must have access to the appropriate initialization parameter files. Oracle Parallel Server uses the initialization parameter files located in *ORACLE_HOME*\DATABASE, unless you specify a different initialization file with the PFILE option at startup.

Editing Initialization Files

To customize Oracle Parallel Server for Windows NT databases functions, you need to edit the initialization parameter files. Use any ASCII text editor to modify the file.

Managing Instances Using OPSM

Although instances can be started and stopped individually from each node using Server Manager, it is not always the most efficient way. OPSM allows you to manager, start and stop one or more instances from one node or from an Oracle Enterprise Manager Console, thus centralizing the management of instances.

Additional Information: See Chapter 6, "Installing and Configuring Oracle Parallel Server Manager", for OPSM requirements and setup.

Using OPSM from the Oracle Enterprise Manager Console

After the repository is built, the Oracle Enterprise Manager Console appears with access to the Oracle Parallel Servers in the Navigator tree. An instance or multiple instances are started or stopped by:

- right-clicking the mouse on a parallel server, and choosing either Startup or Shutdown from the Navigator tree
- executing an Oracle Parallel Server start up/shut down job from the Job menu

---

Additional Information: See Chapter 2, "Oracle Parallel Server Management Operations", in the *Oracle Parallel Server Management User's Guide* and the *Oracle Enterprise Manager Administrator's Guide* for further information about managing Oracle Parallel Servers and parallel server instances from Oracle Enterprise Manager.

If you configured the Oracle Performance Manager, you can display a variety of tabular and graphic performance statistics for parallel server.

Additional Information: See the *Oracle Parallel Server Management User's Guide* for further information about using the Oracle Enterprise Manager Console and Oracle Performance Manager.

Using OPSM from OPSCTL

The OPSCTL.EXE utility allows you to manage all Oracle Parallel Server instances from each node. You can choose to start or stop all instances on an individual node.

Understanding OPSCTL Requirements

Note: The following requirements are met if you followed all the steps in Chapter 5, "Configuring Oracle Parallel Server" and Chapter 6, "Installing and Configuring Oracle Parallel Server Manager".

OPSCTL requires the following:

- The name of each service must be exactly the same as the Oracle SID for that instance (and exactly the same as they appear in the registry).
- The database initialization files must follow a format of INIT*SID*.ORA and be located in the *ORACLE_HOME*\DATABASE directory.
- The database initialization files for all instances of your Oracle Parallel Server must exist on the node where OPSCTL.EXE is run.
- A new Windows NT user with the same name as the one you are using to access your Oracle Enterprise Manager console workstation must be created on all nodes of your Oracle Parallel Server.

  OPSCTL.EXE uses this user to start services on remote machines, so this user must have the equivalent of administrator access on each of the nodes. You can either use the default Oracle Enterprise Manager Console credential to access your Oracle Parallel Server or supply a different account name/password combination when creating a startup/shutdown job. In either case, the account used must have DBA privileges on an instance of Oracle Parallel Server in order to start or stop that instance.

Startup

OPSCTL START Command Line Usage

```
OPSCTL START -CCONNECT_STRING -NDATABASE_NAME  [-ISID, SID] [-F] [-T] [-U] [-M]
[-Y|-E] [-v] [-h]
```

Note: [] indicates optional items.

where:

Parameter	Specifies
-C*CONNECT_STRING*	connect string, such as INTERNAL/MANAGER
-N*DATABASE_NAME*	name of the Oracle Parallel Server database name to start; defined in the INIT_COM.ORA file
-I*SID, SID*	to start specified instance(s) only
-F	startup of instances following an ABORT shutdown; default is IMMEDIATE

Parameter	Specifies
-T	RESTRICTED SESSION privileges only
-U	database not to be mounted upon startup
-M	mount, but do not open the database
-Y	retry database opening if recovery is in progress
-E	EXCLUSIVE mount and open
-V	verbose
-H	print usage

To start all instances:

```
C:\> CD ORACLE_HOME\BIN
C:\ORACLE_HOME\BIN> OPSCTL START -CINTERNAL/PASSWORD -NDATABASE_NAME
```

To start an individual instance:

```
C:\> CD ORACLE_HOME\BIN
C:\ORACLE_HOME\BIN> OPSCTL START -ISID -CINTERNAL/PASSWORD -NDATABASE_NAME
```

Shutdown

OPSCTL STOP Command Line Usage

```
OPSCTL STOP -CCONNECT_STRING -NDATABASE_NAME  [-ISID, SID] [-F] [-T] [-U] [-M]
[-v] [-h]
```

> **Note:** [] indicates optional items.

where:

Parameter	Specifies
-C*CONNECT_STRING*	connect string, such as INTERNAL/MANAGER
-N*DATABASE_NAME*	name of the Oracle Parallel Server database name to stop; defined in the INIT_COM.ORA file
-I*SID, SID*	to stop specified instance(s) only

Parameter	Specifies
-F	shutdown of instances in ABORT mode; default is IMMEDIATE
-T	RESTRICTED SESSION privileges only
-U	database not to be mounted upon startup
-M	NORMAL mode of shutdown; default is IMMEDIATE
-V	verbose
-H	print usage

To stop all instances:

```
C:\> CD ORACLE_HOME\BIN
C:\ORACLE_HOME\BIN> OPSCTL STOP -CINTERNAL/PASSWORD -NDATABASE_NAME
```

To stop an individual instance:

```
C:\> CD ORACLE_HOME\BIN
C:\ORACLE_HOME\BIN> OPSCTL STOP -ISID -CINTERNAL/PASSWORD -NDATABASE_NAME
```

8

Adding Instances and Nodes

This chapter describes how to convert from a single instance Oracle8 database to a multi-instance Oracle8 database using the parallel server option, and how to add a third or fourth node to an existing two or three node configuration. Specific topics discussed are:

- Moving from a Single Instance to Parallel Server
- Adding Additional Nodes to a Cluster

Moving from a Single Instance to Parallel Server

This section explains how to migrate from a single instance Oracle8 database to multi-instance. In addition, it can help you extend an existing OPS configuration to additional nodes. The steps to migrate from a single instance to an Oracle Parallel Server include:

This section should be used as a supplement to the "Migrating from Single Instance to Parallel Server," chapter of the *Oracle8 Parallel Server Concepts and Administration* guide.

Step 1: Configure Hardware

See your vendor documentation for information about setting up Oracle Parallel Server hardware.

Step 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 of, 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.

Step 3: CreateRaw Partitions

When building a database that is to be shared and managed by Oracle Parallel Server, it is necessary to access all disks as raw disks. In the case of a cluster, it is not possible to share a file system between two separate nodes. The file system is in effect a private cache resident on just one node.

The process of creating a raw disk involves creating an extended partition on a disk and assigning logical partitions to it.

To create raw partitions:

1. Create raw partitions and assign logical partitions, as described in Chapter 3, "Performing Pre-Installation Tasks". Ensure enough raw partitions have been created for:

File Type	Number of Partitions	Space Per Partition
control files	2	2 MB
log files[1]	2 per node	small number of transactions: 20-50K each
		medium number of transaction: 200-500 K each
		large number of transactions: 2 MB - 5 MB each

[1] Each node requires at least two log files.

2. Assign meaningful symbolic link names to the logical partitions, as described in "Step 2: Assign Symbolic Links to Each Logical Partition" on page 5-2. You may need to modify the ORALINK*.TBL files to suit your database.

Step 4: Install Oracle8 Enterprise Edition and Oracle Parallel Server Option

Except on the server already running, install Oracle8 Enterprise Edition along with the Oracle Parallel Server option, as described in Chapter 4, "Installing Oracle Parallel Server". If your current single instance database does not have the supporting hardware, you must perform a clean install on new cluster hardware.

On the server already running Oracle8 Enterprise Edition, install the Oracle Parallel Server Option, as described in Chapter 4, "Installing Oracle Parallel Server".

Step 5: Install OSD Files

For all nodes, see your Operating System Dependent (OSD) vendor documentation for instructions about installing OSD files.

Step 6: Move Initialization Files

The initialization files, INIT_COM.ORA and INIT*SID*.ORA are installed in the *ORACLE_HOME*\OPS directory. In order to avoid having to specify the PFILE parameter when starting the database (STARTUP PFILE=C:ORANT\OPS\INIT*SID*.ORA), move these file to the *ORACLE_HOME*\DATABASE file where initialization files are normally located.

To change the location of the initialization files:

1. Move INIT_COM.ORA and INIT*SID*.ORA form the *ORACLE_HOME*\OPS directory to *ORACLE_HOME*\DATABASE directory.
2. Modify the INIT*SID*.ORA file so the IFILE points to the new *ORACLE_HOME*\DATABASE directory.

Step 7: Export Data from Old Database

Export the entire database from the single instance database. Use one of the following tools:

- Oracle Data Manager
- Export Utility

Oracle Data Manager

To start Oracle Data Manager from the Enterprise Manager Console:

- Choose View > Launch Palettes > Applications.

  The Applications palette appears. Click the Data Manager icon.

- Choose Tools > Applications > Data Manager

Additional Information: See the *Oracle Enterprise Manager Administrator's Guide*.

Export Utility

Enter the following at the MS-DOS command prompt followed by your user name and password.

> **Note:** To export an entire database, you must use the user name SYSTEM. Do not use INTERNAL or SYS.

To use the Export utility to export all data from an existing database to the new database:

1. Set ORACLE_SID to the database service of the database whose contents you want to export. For example, export the starter database ORCL. Note there are no spaces around the equal sign (=) character.

   ```
   C:\MYDIR> SET ORACLE_SID=ORCL
   ```

2. Run the Export utility:

   ```
   C:\MYDIR> EXP80 SYSTEM/PASSWORD FILE=MYEXP.DMP FULL=Y LOG=MYEXP.LOG
   ```

   You now have a full database export of the starter database ORCL in the file MYEXP.DMP, with all messages from the Export utility logged in the MYEXP.LOG file.

   When running the Export utility, the default values for the following parameters under Windows NT are:

   BUFFER 4 KB

   RECORDLENGTH 2 KB

Step 8: Prepare CREATE DATABASE Script

A new database must be created on the raw partitions.

To prepare the CREATE DATABASE script:

1. Use the BUILD_DB.SQL script located in *ORACLE_HOME*\RDBMS80\ADMIN to build a script:

   Make the following changes to the BUILD_DB.SQL script:

   a. Set PFILE so it points to the *ORACLE_HOME*\DATABASE\INITOPS1.ORA initialization file.

   b. Modify the CREATE DATABASE ORACLE to CREATE DATABASE OPS.

c. Modify the log file and data file names with the symbolic link names you created in "Step 2: Assign Symbolic Links to Each Logical Partition" on page 5-2.

d. Modify the log file and data files with a naming convention of \\.*SYMBOLIC_LINK_NAME*. The symbolic link name should match the name you used in ORALINK*.TBL file, as described in "Step 3: CreateRaw Partitions" on page 8-3.

e. Create new entries for additional log files for each additional node. (Each node requires two log files.

f. Modify the log file and data file sizes for the Oracle Parallel Server.

Enter the following command to find out the current size of data files:

```
SELECT * FROM DBA_DATA_FILES
```

g. Create enough private (acquired explicitly by an instance when an instance opens a database) rollback segments (16 is the default) for the number of concurrent users per transaction. With the exception of the SYSTEM rollback segment, public rollback segments cannot be shared among nodes.

A sample script follows:

```
create database ops
    controlfile reuse
logfile GROUP 1 '\\.\OPS_log1t1'size 200K reuse,
            GROUP 1 '\\.\OPS_log2t1'size 200K reuse,
            GROUP 2 '\\.\OPS_log1t2'size 200K reuse,
            GROUP 2 '\\.\OPS_log2t2'size 200K reuse,
    datafile '\\.\OPS_sys01' size 50M,
    character set WE8ISO8859P1;

create rollback segment rb_temp;

create tablespace user_data
    datafile '\\.\OPS_usr01' size 15M;
create tablespace rollback_data
    datafile '\\.\OPS_rbs01' size 50M;
create tablespace temporary_data
    datafile '\\.\OPS_tmp01' size 10M;
alter rollback segment rb_temp online;

-- Change the SYSTEM users' password, default tablespace and
-- temporary tablespace.
```

```
alter user system temporary tablespace temporary_data;
alter user system default tablespace user_data;

-- Create 16 rollback segments.  Allows 16 concurrent users with open
-- transactions updating the database. This should be enough.

create private rollback segment rb1 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb2 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb3 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb4 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb5 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb6 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb7 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb8 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb9 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb10 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb11 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb12 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb13 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb14 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb15 storage(initial 50K next 50K)
  tablespace rollback_data;
create private rollback segment rb16 storage(initial 50K next 50K)
  tablespace rollback_data;
```

Step 9: Configure OSD Layer, Create Services, and Configure Network

Follow "Step 3: Configure OSD Layer" on page 5-7 to "Step 6: Configure the Network" on page 5-12.

Step 10: Create an Oracle Parallel Server Database

To create a database:

1. On the *primary* node, start the database, and run the CREATE DATABASE script you just created along with the following scripts:

CATALOG.SQL	Generates the data dictionary.
CATPROC.SQL	Installs the objects used by the Oracle7 database's PL/SQL functionality.
CATPARR.SQL	Creates objects for Oracle Parallel Server.

```
C:\> SVRMGR30
SVRMGR> CONNECT INTERNAL/ORACLE
SVRMGR> STARTUP
SVRMGR> @%ORACLE_HOME%\RDBMS80\ADMIN\CREATE_DATABASE.SQL
SVRMGR> @%ORACLE_HOME%\RDBMS80\ADMIN\CATALOG.SQL
SVRMGR> @%ORACLE_HOME%\RDBMS80\ADMIN\CATPROC.SQL
SVRMGR> @%ORACLE_HOME%\RDBMS80\ADMIN\CATPARR.SQL
```

2. If you modified the number of rollback segments, change the *ORACLE_HOME*\DATABASE\INIT*SID*.ORA files to reflect the change.

3. Shut down the database.

```
SVRMGR> SHUTDOWN
```

Step 11: Transfer Data From Old Data to Empty Database

Import the entire database into the empty database. Use one of the following tools:

- Oracle Data Manager
- Import Utility

Oracle Data Manager

To start Oracle Data Manager from the Enterprise Manager Console:

- Choose View > Launch Palettes > Applications. The Applications palette appears. Click the Data Manager icon.
- Choose Tools > Applications > Data Manager

Additional Information: See *Oracle Enterprise Manager Administrator's Guide*.

Import Utility

To import the full export created in the section "Step 7: Export Data from Old Database" on page 8-4 into the new database:

```
C:\MYDIR> IMP80 SYSTEM/PASSWORD FILE=MYEXP.DMP FULL=Y LOG=MYIMP.LOG
```

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.

When running the Import utility, the default values for the following parameters under Windows NT are:

```
BUFFER 4 KB
RECORDLENGTH 2 KB
```

Step 12: Start the Database

See "Step 8: Start the Database in Parallel Mode" on page 5-17 to start the Oracle Parallel Server.

Adding Additional Nodes to a Cluster

This release of Oracle Parallel Server supports up to four nodes in a cluster. If you have already configured a two- or three-node cluster and want to add nodes three and / or four, you must add the new configuration information for these nodes through scripts. The database does need to be recreated.

Step:		For additional information, see:
1.	**Create additional logical partitions.** Each additional node requires two log files. Because of this, you must add two additional logical partitions for each additional node.	"Step 2: Create Logical Partitions In an Extended Partition" on page 3-9
2.	**Unassign drive letters.**	"Step 3: Unassign Drive Letters" on page 3-13
3.	**Assign symbolic links to each additional logical partitions.**	"Assign Symbolic Links to Each Logical Partition" on page 8-11
4.	**Shut down services.**	"Shut Down Services" on page 8-12
5.	**Configure OSD layer.**	"Step 3: Configure OSD Layer" on page 5-7
6.	**Create OracleService*SID* services.**	"Step 4: Create Services" on page 5-8
7.	**Add the ORACLE_SID entry to the registry.**	"Step 5: Add the ORACLE_SID Entry to the Registry" on page 5-10
8.	**Configure the network.** With the addition of nodes, additional service names must be created in order for Net8 clients to communicate with Oracle8 servers.	"Step 6: Configure the Network" on page 5-12
9.	**Create additional rollback segments for new nodes.**	"Create Rollback Segments For Additional Nodes" on page 8-13
10.	**Start the database.**	"Step 8: Start the Database in Parallel Mode" on page 5-17
11.	**Verify instances are running.**	"Step 9: Verify InstancesAre Running" on page 5-19
12.	**Reconfigure Performance Manager.**	"Step 3: Configure Oracle Performance Manager" on page 6-18.

Assign Symbolic Links to Each Logical Partition

Use the SETLINKS utility to create symbolic links to the new logical partitions. The SETLINKS utility uses a file named ORALINK2.TBL for the third node in a cluster and ORALINK3.TBL for the fourth node in a cluster. These files will create the symbolic links to the log files' logical partitions:

Figure 8–1 ORALINK2.TBL

```
OPS_log1t3 \Device\Harddiskx\Partitionx
OPS_log2t3 \Device\Harddiskx\Partitionx
```

Figure 8–2 ORALINK3.TBL

```
OPS_log1t4 \Device\Harddiskx\Partitionx
OPS_log2t4 \Device\Harddiskx\Partitionx
```

To create symbolic links to a raw partition:

1. On the *primary* node, edit the partition and hard disk numbers appropriately in the ORALINK2.TBL file (for the third node) and ORALINK3.TBL (for the fourth node) located in *ORACLE_HOME*\OPS to match the disk numbers and partitions number you wrote down in section Step 2: Unassign Drive Letters.

   You can use a worksheet similar the one below to assist with the edits.

Symbolic Link	Node 3
OPS_log1t3	Harddisk*x* Partition*x*
OPS_log2t3	Harddisk*x* Partition*x*

Symbolic Link	Node 4
OPS_log1t4	Harddisk*x* Partition*x*
OPS_log2t4	Harddisk*x* Partition*x*

WARNING: Do *not* change the symbolic link names in OPSLINK1.TBL. If you do, symbolic link names must be modified for the following files located in:

- ***ORACLE_HOME*\OPS\OPS.SQL**
- ***ORACLE_HOME*\DATABASE\INIT_COM.ORA**

2. Run the ORALINK2.TBL and/or ORALINK3.TBL through the SETLINKS application as shown below:

```
C:\> CD ORACLE_HOME\OPS
C:\ORACLE_HOME\OPS> SETLINKS /F:ORALINK2.TBL
C:\ORACLE_HOME\OPS> SETLINKS /F:ORALINK3.TBL
```

   The symbolic links are created:

```
Oracle Corporation. Copyright (c) 1997. All rights reserved.
Created Link:OPS_log1t3 = Device:\Device\Harddisk3\Partition15
Created Link:OPS_log2t3 = Device:\Device\Harddisk3\Partition16
Dos devices updated successfully.
```

3. Make sure the drives have been mapped with the correct names with the SETLINKS application (as shown below):

```
C:\ORACLE_HOME\OPS> SETLINKS /D
```

4. Repeat Steps 1-3 on the other nodes.

Shut Down Services

Some vendors may require you stop the services and stop the Cluster Manager prior to configuring the OSD layer. See your vendor documentation for further information.

To stop the services:

Stop OracleService*SID*, Oracle*HOME_NAME*TNSListener80, OraclePGMSService, and OracleAgent services on each node:

- From the MS-DOS command line, enter:

```
C:\> NET STOP SERVICE
```

- From the Control Panel's *Services* window, select the service, and click Stop.

Create Rollback Segments For Additional Nodes

To create rollback segment for the additional nodes:

1. On *each* node, delete the *ORACLE_HOME*\RDBMS80\TRACE*SID*LMON.TRC file.
2. Start OracleService*SID* on *each* node.

   Note: Prior to starting a OracleService*SID* service, the OraclePGMSService service must be running. If you used the CRTSRV script in "Step 4: Create Services" on page 5-8, OraclePGMSService automatically starts when the OracleService*SID* service is started.

   If you chose to create your services with another method, you can still have OraclePGMSService start up automatically with a OracleService*SID* service by entering the following at the command for each node:

   ```
   C:\> OPSREG80 ADD SID
   ```

   - From the MS-DOS command line, enter:

     ```
     C:\> NET START OracleServiceSID
     ```

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

3. On the *primary* node, start the database, and run the C_OPS3.SQL (for the third node) and / or C_OPS4.SQL (for the fourth node) scripts:

   ```
   C:\> SVRMGR30
   SVRMGR> CONNECT INTERNAL/PASSWORD
   SVRMGR> STARTUP
   SVRMGR> @C_OPSX.SQL
   ```

   The C_OPS*X*.SQL script creates the necessary rollback segments for the third and fourth nodes.

4. On *each* node, ensure there is a "*Reconfiguration complete*" message in the *ORACLE_HOME*\RDBMS80\TRACE*SID*LMON.TRC file for the correct number of node, signifying the cluster was started without errors:
5. Stop OracleService*SID*:

   - From the MS-DOS command line, enter:

```
C:\> NET STOP OracleService SID
```

- From the Control Panel's *Services* window, select the service, and click Stop.

9

Backing Up and Recovering

This chapter is intended as an add-on to Chapter 13, “Backup and Recovering Database Files”, of the *Oracle8 Enterprise Edition Getting Started*, which describes how to back up and recover an Oracle8 Enterprise Edition database. Specific topic covered in this chapter is:

- Backup and Recovery

Backup and Recovery

Backup and recovery of an Oracle Parallel Server database is performed from one node of the cluster using the NT Backup Manager and the NT Recovery Manager.

These applications operate in the same way for Oracle Parallel Server as Oracle8 Enterprise Edition, except where noted below:

- All instances in the cluster must be shut down before performing an off-line backup or recovery.
- 4mm and 8mm tapes are supported for Oracle Parallel Server
- The ORACLE_SID parameter must be set in the registry. See "Step 5: Add the ORACLE_SID Entry to the Registry" on page 5-10.
- The NT Recovery Manager requires the INIT*SID*.ORA to be located in the *ORACLE_HOME*\DATABASE directory. INIT*SID*.ORA is placed in *ORACLE_HOME*\OPS by default. To use the NT Recovery Manager, copy INIT*SID*.ORA to *ORACLE_HOME*\DATABASE.
- When performing an off-line backup, you must change the default file names to the symbolic link names, \\.*SYMBOLIC_LINK NAME*.

  For example, you would click Files from the *Oracle Backup Manager* dialog box:

The *Database Files* dialog box appears. Select each file type (Data Files, Log Files and Control Files), and enter the appropriate file name. File names should match the symbolic link names you used in the ORALINK*x*.TBL file(s):

For additional information, see "Step 2: Assign Symbolic Links to Each Logical Partition" on page 5-2.

A

Directory Structure

Specific topic covered in this appendix is:

- Oracle Parallel Server Directory Structure

Oracle Parallel Server Directory Structure

The Oracle Installer places all Oracle products for Windows NT into subdirectories of *ORACLE_HOME*. The actual directory names vary according to the different products you install.

The subdirectories of *ORACLE_HOME* contain the contents listed below.

All .DLL and .EXE files and configuration variables have their version numbers appended. This convention allows you to maintain multiple releases of these items for backward compatibility. For example, the executable for the Oracle Names Control Utility version 8.0 is NAMESCTL80.EXE.

Directory Name	Contents
\A2OWIZ80	Oracle Migration Assistant for Microsoft Access files
\AGENTBIN	Oracle Intelligent Agent files and subdirectories
\BIN	executables, help files, and library files
\CORE40	Oracle8 database message files
\DATABASE	database, password, and initialization files
\DBS	Oracle Toolkit II message files, SQL scripts and other utility files
\EERN806	Oracle8 Enterprise Edition Release Notes
\JRE11	files and subdirectories that allows Java applications to run
\MSHELP	Windows help files
\NET80	message files and subdirectories for Net8 products **Additional Information:** See *Net8 Getting Started for Windows NT and Windows 95* for additional information on the \NET8 directory and its subdirectories.
\NLSRTL33	NLS message files
\OPS	Oracle Parallel Server SQL scripts
\ORAINST	Oracle Installer files
\OTRACE80	Oracle Trace files. \OTRACE80 contains the following subdirectories:

Directory Name	Contents
\PLSQL80	message files, SQL scripts, and demonstration files for PL/SQL
\PLUS80	SQL*Plus SQL and message files
\RDBMS80	Oracle8 Enterprise Edition message, resource, and README files
\SYSMAN	Oracle Enterprise Manager files and subdirectories
\VS10	NT Backup Manager and NT Recovery Manager files

B

Troubleshooting

Specific topics covered in this appendix are:

- Cluster Configuration Tips
- CM Troubleshooting
- Performance and Manager Configuration Tips
- Starting Services
- DYNAMIC RESOURCES ALLOCATED or DYNAMIC LOCKS ALLOCATED
- Understanding the Trace Files
- Cluster Tracing
- Using PhysicalDrive for Raw Partitions
- SHUTDOWN ABORT
- Contacting Oracle Worldwide Customer Support

Cluster Configuration Tips

A large fraction of cluster problems that have been reported to Oracle Corporation are due to incorrect cluster configuration, particular of the Cluster Manager (CM) and interconnect components.

The information in this section is based on Oracle Corporation's reference implementation of the cluster Operating System Dependent (OSD) modules. Consequently, some of this information may not be applicable to your particular cluster environment.

Additional Information: Consult with your hardware vendor for more details about installing and configuring your particular cluster configuration

---

Note: The registry instructions in this section assume REGEDT32, not REGEDIT.

Cluster Software

Make sure all nodes have the exact same cluster OSD software installed, as well as the same registry configuration. Software can be verified by ensuring nodes have the same time stamps and file sizes.

CM Configuration

Typically, each node in a cluster will have at least two cards, one for the corporate network and one for the cluster interconnect. A computer, however, can only have one host name associated with it. To get around this problem, a host name for the computer can be assigned just for the cluster interconnect.

To specify a host name for the cluster interconnect:

1. For each node, ping the host name. For example,

```
C:\> PING OPS1-NT.US.ORACLE.COM

A message similar to the one below appears:
Reply from 144.25.188.247: bytes=32 time<10ms TIL=126
```

The IP address returned is for the corporate network, not the cluster interconnect.

2. For each node, determine which ethernet card will be used for the cluster interconnect by entering:

```
C:\> IPCONFIG /ALL
```

The output looks similar to the sample shown below:

```
Windows NT IP Configuration

            Host Name . . . . . . . . . . : ops1-nt.us.oracle.com

Ethernet adapter El90x1:

            Description . . . . . . . . . : 3Com 3C90x Ethernet Adapter
            IP Address. . . . . . . . . . : 144.25.188.247

Ethernet adapter CpqNF31:

            Description . . . . . . . . . : Compaq NetFlex-3 Driver
            IP Address. . . . . . . . . . : 144.25.190.247
```

In this case, the first interface is used for the corporate network, while the second interface is (144.25.190.247) is the one intended for the cluster interconnect.

3. Specify an new host names for each node's interconnect IP address in the HOST file (*SYSTEMROOT*\SYSTEM32\DRIVERS\ETC\HOSTS). For example:

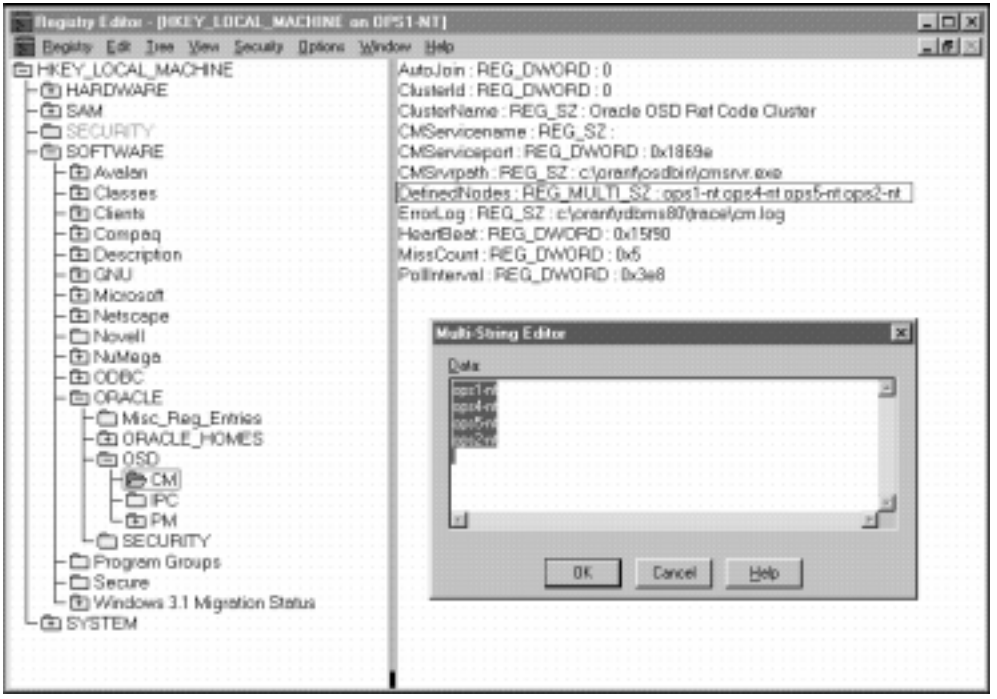
```
144.25.190.247 ops1-ipc
144.25.190.248 ops2-ipc
144.25.190.249 ops3-ipc
144.25.190.250 ops4-ipc
```

The HOSTS file should have one entry for each node's interconnect, and should be copied to all nodes of the cluster so that they can see each other. To verify that they can see each other, try pinging each host from each node. For example

```
C:\> PING OPS3-IPC
```

4. For each node, ensure the DefinedNodes value is specified in HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD>CM. DefinedNodes specifies the member nodes in the cluster.

```
DefinedNodes: REG_MULTI_SZ: ops1-ipc ops4-ipc ops5-ipc ops2-ipc
```



Note: DefinedNodes must be of value class REG_MULTI_SZ, and each host name entry must be entered on a separate line in the *Multi-String Editor* dialog box.

---

5. For each node, ensure the CmHostName value is specified in HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD>CM. CmHostName specifies the node's interconnect host name.

```
CmHostName: REG_SZ: ops1-ipc
```

Cluster Configuration Verification

To verify your cluster configuration:

1. Start PGMS each node:
   - From the MS-DOS command line, enter:

     ```
     C:\> NET START ORACLEPGMSSERVICESID
     ```

   - From the Control Panel's *Services* window, select OraclePGMSService, and click Start.
2. Check the bottom of PGMS.LOG file stored in *SYSTEMROOT*\SYSTEM32\PGMS.LOG to ensure that each time a node is brought up, PGMS reconfigures with the correct number of nodes. For example, if two nodes are up, the following should be in the log file:

   ```
   15:06:46 | MESSAGE | 006f | HandleReconfig(): Reconfig OK - nodes(2)
   rcfgGen(5) master(0)
   ```

3. If you are unable to bring up PGMS, check your cluster configuration to make sure that it is correct.

CM Troubleshooting

During normal operation, CM on each node checks in with one another to ensure the health of each member. These check-ins occur at interval of *N* in milliseconds, as specified by the PollInterval registry value in HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD>CM. A node is allowed to miss *M* check-ins before it is cast out of the cluster, as specified by the MissCount value in HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD>CM.

Failed check-ins are recorded to the CM error log file (CM.LOG). These check-in packets are typically UDP packets, and may be lost:

- under heavy activity
- due to network congestion (if there is not a dedicated interconnect)

If one of your database instances is dropping out of the cluster under heavy activity, you may see messages in CM.LOG file similar to:

```
05:01:25 | MESSAGE | PollingThread(): node(1) missed(3) checkin(s)
05:01:27 | MESSAGE | PollingThread(): node(1) missed(5) checkin(s)
05:01:28 | MESSAGE | PollingThread(): node(1) failure detected
```

This occurs if the check-in messages were lost because of the heavy activity. Make sure there is a dedicated interconnect for Oracle Parallel Server that is separate from the rest of the network. Slightly increasing the MissCount value may also help.

Note: MissCount * PollInterval should never be greater than 20 seconds.

CM Secondary Backup

If you are using the secondary disk backup feature of the CM, try to use a partition on a disk that is not heavily used. The backup disk file is written to by every node member during each check-in. If the backup disk is heavily used, it may cause the CM to miss check-ins and falsely drop node members.

Note: If you are using the secondary disk backup feature, do not lower PollInterval beyond 500 milliseconds because every node writes to the disk backup partition every PollInterval.

CM Error Log File Specification

The CM error log file (CM.LOG) is specified by the ErrorLog value in HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD>CM:

```
ErrorLog: REG_SZ: c:\orant\rdbms80\trace\cm.log
```

Oracle Corporation recommends specifying an error log location of ORANT\RDBMS80\TRACE\CM.LOG.

Performance and Manager Configuration Tips

You must configure the Performance and Management (PM) module so that PGMS can determine the cluster configuration. Each Oracle Parallel Server database corresponds to a PGMS group or domain. For example, the INIT*SID*.ORA and INIT_COM.ORA files could have the following parameters defined:

INITOPS1.ORA:

```
instance_number=1
```

INITOPS2.ORA:

```
instance_number=2
```

INITOPS3.ORA:

```
instance_number=4
```

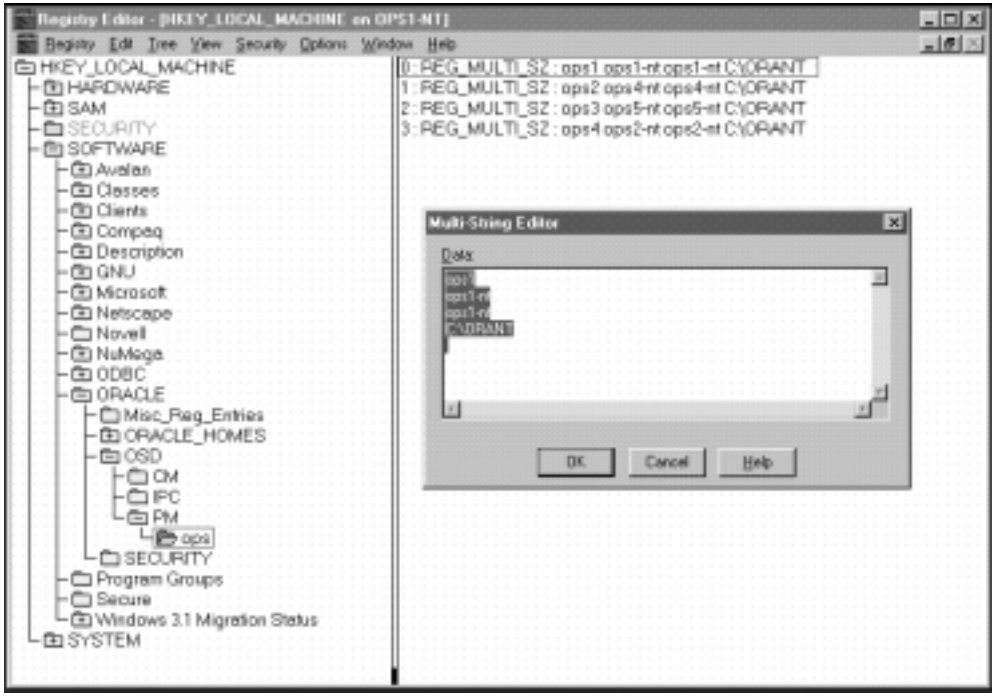
INITOPS4.ORA:

```
instance_number=4
```

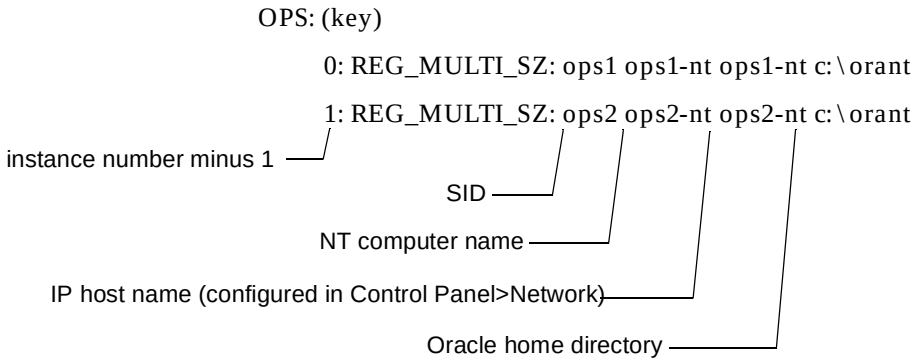
INIT_COM.ORA:

```
db_name=ops
```

The HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD>PM key would then contain:



where:



Note: Each row entry must be entered on a separate line in the *Multi-String Editor* dialog box. Instance numbers must be sequential, such as 0, 1, 2. Do not skip instance numbers, such as 0, 1, 3. Also, the key name (OPS) must match the value of DB_NAME in INIT_COM.ORA

ORA-29702

If the instance numbers in the PM key do not match those specified in the INIT*SID*.ORA file, you will receive the following error in *ORACLE_HOME*\RDBMS80\TRACE*SID*LMON.TRC upon instance startup:

```
ORA-29702: error occurred in Group Membership Service operation
```

Starting Services

If you are having difficulty starting services or the database, check the PGMS.LOG file stored in *SYSTEMROOT*\SYSTEM32\PGMS.LOG.

If you used the CRTSRV script in "Step 4: Create Services" on page 5-8, OraclePGMSService automatically starts up and shuts down when the OracleService*SID* service is started.

If you did not use the CRTSRV script, you can still have OraclePGMSService start up automatically with a OracleService*SID* service by entering the following at the command for each node:

```
C:\> OPSREG80 ADD SID
```

You can also discontinue the OraclePGMSService service automatic start up with OracleService*SID* service with the following at the command line for each node:

```
C:\> OPSREG80 DEL SID
```

DYNAMIC RESOURCES ALLOCATED or DYNAMIC LOCKS ALLOCATED

The following messages appear if LM_RESS and LM_LOCKS values are not sufficient, and additional IDLM locks or resources must be allocated dynamically from the SGA:

- `DYNAMIC RESOURCES ALLOCATED`
- `DYNAMIC LOCKS ALLOCATED`

If these messages appear often, it may lead to SGA exhaustion. To resolve this, increase LM_RESS and LM_LOCKS parameters appropriately based on your database needs to avoid exhausting the SGA.

Additional Information: See Chapter 15, "Allocating PCM Instance Locks Oracle Parallel Server," of the *Oracle8 Parallel Server Concepts and Administration* guide.

Understanding the Trace Files

This section discusses the following trace file subjects:

- Background Thread Trace Files
- User Thread Trace Files
- Alert File
- Error Call Trace Stack
- PGMS Tracing
- CM Tracing

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_COM.ORA initialization parameter file. If BACKGROUND_DUMP_DEST is set, the trace files are stored in the directory specified. If the parameter is not set, the trace files are stored in the *ORACLE_HOME*\RDBMS80\TRACE directory.

Oracle8 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*LCK*n*.TRC	Trace file for the LCK*n* process. This trace file shows lock request for other background processes.
*SID*LMD*n*.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 show status of cluster, including the "Reconfiguration complete" message.
*SID*P00*n*.TRC	Trace file for the parallel query slaves.

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 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_COM.ORA initialization parameter file. If the

BACKGROUND_DUMP_DEST parameter is not set, the *SID*ALRT.LOG file is generated in *ORACLE_HOME*\RDBMS80\TRACE.

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 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:

```
C:\> SVRMGR30
SVRMGR30> CONNECT INTERNAL/PASSWORD
SELECT PID "Oracle Process Id",
       NAME
    FROM V$PROCESS, V$BGPROCESS
    WHERE V$PROCESS.ADDR = V$BGPROCESS.PADDR;
```

   Output displayed looks like this:

```
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 out the trace stack of LMON, enter:

   a. Set the Oracle process ID to LMON, which is 3 in this example:

```
SVRMGR30> ORADEBUG SETORAPID 3
```

   b. Dump the error stack to *SID*LMON.TRC:

```
SVRMGR30> ORADEBUG DUMP ERRORSTACK 3
```

Cluster Tracing

CM and PGMS tracing can be helpful to Oracle Worldwide Support in debugging your cluster configuration problems in cases where the database is not starting, a particular node is hanging, or there is a node crash.

PGMS Tracing

PGMS tracing is stored in the PGMS log file, *SYSTEMROOT*\SYSTEM32\PGMS.LOG.

Note: Do not enable detailed tracing during normal database operation.

To enable detailed PGMS tracing:

1. De-install the OraclePGMSService:

   ```
   PGMS /R
   ```

2. Re-install OraclePGMSService with debug flags turned on:

   ```
   PGMS /I:"C:ORANT\BIN\PGMS.EXE /D /V /S"
   ```

   where:

/D	debug tracing
/V	verbose tracing
/S	spy on PGMS network packets

To disable tracing:

1. De-install the OraclePGMSService:

   ```
   PGMS /R
   ```

2. Re-install OraclePGMSService with debug flags turned off:

   ```
   PGMS /I:C:"ORANT\BIN\PGMS.EXE"
   ```

CM Tracing

CM tracing is stored in the error log file, CM.LOG. The location of CM.LOG is defined by the ErrorLog value in HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD>CM.

To enable detailed CM tracing:

1. Stop the CMSRVR.EXE by rebooting the node.
2. Specify the CMSrvrpath value in HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD>CM. ErrorLog specifies the CM log file.

   ```
   CMSrvrpath: REG_SZ: c:\orant\osdbin\cmsrvr.exe /v /c /s
   ```

   where:

/v	verbose
/c	trace client request
/s	spy on CM network traffic

Using PhysicalDrive for Raw Partitions

When creating symbolic links for the logical partitions with SETLINKS utility, do not use prefix \\.\PhysicalDrive. If you use \\.\PhysicalDrive as a symbolic link, you may corrupt the database files. Use the symbolic links provided in the ORALINK*x*.TBL files, described in Chapter 5, "Configuring Oracle Parallel Server".

SHUTDOWN ABORT

SHUTDOWN ABORT is not recommended. Oracle Corporation recommends shutting down the OracleService*SID* service so that resources, such as memory usage or files, will be cleaned up by the Windows NT operating system correctly.

To shut down OracleService*SID*:

- From the MS-DOS command line, enter:

  ```
  C:\> NET STOP OracleServiceSID
  ```

- From the Control Panel's *Services* window, select the OracleService*SID* service, then choose Stop.

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 at hand:

- cluster hardware, for example, a two-node cluster of Dell PowerEdge 6100 servers
- Windows NT version (for example, Windows NT (Workstation, Server, Enterprise) 4.0 with Service pack 3)
- all five digits in release number of Oracle RDBMS (for example, 8.0.5.1.0)
- all five digits in release number of Oracle Parallel Server Option
- version number of PGMS, which can be obtained from *SYSTEMROOT*\SYSTEM32\PGMS.LOG.
- contents of HKEY_LOCAL_MACHINE>SOFTWARE>ORACLE>OSD key
- cluster OSD upgrades from vendor
- particular operation that failed, for example, database startup or query
- steps to reproduce the problem.

Severe Errors

If an ORA-600 error occurred, it will be printed to *SID*ALRT.LOG file. If an ORA-600 error or any other severe errors appear in the *SID*ALRT.LOG file, then provide all files in *ORACLE_HOME*\RDBMS80\TRACE and PGMS.LOG located in *SYSTEMROOT*\SYSTEM32.

Glossary

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

Operating System Dependent component that discovers and tracks the membership state of nodes by providing a common view of cluster membership across the cluster.

Also referred to as CM.

CM

An abbreviation for Cluster Manager.

DLL

Dynamic Link Libraries. Microsoft Windows operating system executable files.

domain

A set of instances coordinating together that run on only the nodes defined within a cluster. Typically, a domain is comprised of instances that run on some subset of those nodes. Multiple domains can exist with a cluster. Each domain defines the Oracle Parallel Server instance. Each domain name is typically the database name defined in the INIT_COM.ORA file.

extended partition

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

avoids the four-partition limit by allowing you to define large numbers of logical partitions to accommodate applications using Oracle8 Server.

fine-grain locks

A type of locking where PCM locks are dynamically allocated at block-access time. The resources for the lock are only allocated during the time the lock is needed and are released when the lock is released.

Group Membership Service

A service called OraclePGMSService that monitors what groups (or domains) are up and its instance members. OraclePGMSService interacts with the Cluster Manager — the vendor software that manages access to shared disks and monitors the status of various cluster resources (including nodes, networks, and the PGMS). The Cluster Manager provides a node monitor service, which each PGMS uses to determine the status of PGMS instances on other nodes.

Also referred to as PGMS.

hashed locks

A type of locks where PCM locks are statically assigned to one or more blocks in the data files.

IDLM

An abbreviation for the Integrated Distributed Lock Manager.

IO

An abbreviation for Input/Output.

Input/Output

Operating System Dependent component that provides I/O to access shared disks.

Also referred to as IO.

instance

A set of threads and resources required for proper database operation. The pool of threads that make up an instance coordinate their work on the database through shared memory by means of the Shared Global Area (SGA). All processes belonging to the instance exist on the same node. You can connect to any instance to access any information that resides within a parallel server database.

Integrated Distributed Lock Manager

Oracle Parallel Server software that provides locking mechanisms to control allocation and modification of Oracle resources.

Also referred to as the IDLM.

Interprocess Communication

Operating System Dependent component that reliable transfer of messages between instances on different nodes.

Also referred to as IPC.

IPC

An abbreviation for Interprocess Communication.

LMD*n* process

Process that handles remote lock requests (those which originate from another instance).

LMON process

Process that manages instance and processes deaths and associated recovery for the IDLM.

listener

The server process that listens for and accepts incoming connection requests from client applications. Oracle listener processes start up Oracle database processes to handle subsequent communications with the client.

LISTENER.ORA

A configuration file that describes one or more TNS listeners on a server.

logical partition

Raw partition created by the Disk Administrator that points to a partition other than Partition0.

node

A Windows NT server where an instance resides.

OPSCONF

A utility to generate configuration files and a script needed for the Oracle Performance Manager.

OPSCTL Utility

OPSM uses the OPSCTL utility (installed on each node) to manage instances. This utility is run directly from the command line on a node, or called by the Oracle Enterprise Manager Console. OPSCTL gathers information about all the instances in a cluster from the TNSNAMES.ORA files.

OPSM

An abbreviation for Oracle Parallel Server Management.

Oracle Enterprise Manager

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

Oracle Net8 Easy Config

A GUI application to generate a TNSNAMES.ORA file.

Oracle Parallel Query

Parallel query divides the work of processing certain types of SQL statements among multiple query server processes.

Oracle Parallel Server

An architecture that allows multiple instances to access a shared database. Oracle Parallel Server offers the following (terms will be described later in this chapter):

Oracle Parallel Server Manager

A comprehensive and integrated system management solution for the Oracle Parallel Server. Oracle Parallel Server Manager allows you to manage multi-instance databases running in heterogeneous environments through an open client-server architecture.

Also referred to as OPSM.

Oracle Parallel Server Manager

A single point for starting, stopping, and monitoring the activity of parallel servers and parallel server instances from the command line or from within Oracle Enterprise Manager.

Oracle Performance Manager

A tools that offers a variety of tabular and graphic performance statistics for parallel servers. The statistics represent the aggregate performance for all instances running on a parallel server.

Oracle service

Created and associated with Oracle8 database. Similar to Windows NT services.

Oracle8 database

The software used to create and maintain the database system, as well as the actual data stored in the database.

Oracle8 ORDBMS

The Oracle Object Relational Database Management System. Oracle Parallel Server for Windows NT is an Oracle RDBMS.

Operating System Dependent layer

A software layer that consists of several software components developed by vendors. The OSD layer maps the key OS/cluster-ware services required for proper operation of Oracle Parallel Server.

OPQ

An abbreviation for Oracle Parallel Query.

OPSCONF

A utility to generate configuration files and a database link SQL script needed by the Oracle Performance Manager.

OPSCTL

A utility to start and stop instances from either one node or from Oracle Enterprise Manager.

OSD

An abbreviation for Operating System Dependent.

P&M

An abbreviation for Performance and Management.

Parallel Cache Management

Oracle Parallel Server software that provides instance locks that cover one or more data blocks (table or index blocks) in the buffer cache. PCM locks ensure that each shared buffer cache in a node remains consistent with the shared buffer caches in other nodes with minimal use of the DLM.

Also referred to as PCM.

parallel coordinator

A process that dispatches the execution of a SQL statement to several *parallel server* processes and coordinates the results from all of the slave processes to send the results back to the user.

parallel server process

Processes that performs the requested action of the parallel coordinator.

Performance and Management

Operating System Dependent component that supports external performance and management tools.

PCM

An abbreviation for Parallel Cache Management.

PGMS

An abbreviation of Group Membership Service.

physical disk

Partition that points to Partition0. Window NT automatically creates a symbolic link name of \\.\PhysicalDrive*x* (where *x* is the physical drive number) to \Device\Harddisk*x*\Partition0. The *x* in \\.\PhysicalDrive*x* matches the *x* in \Device\Harddisk*x*\Partition0.

primary partition

A partition into which you can install the files needed to load an operating system. A primary partition is usually formatted for a particular file system and is assigned a drive letter. There can be at most four primary partitions on a physical drive.

SID

A unique name for an Oracle database instance. To switch between Oracle databases, users must specify the desired SID. The SID is included in the

CONNECT DATA parts of the connect descriptors in a TNSNAMES.ORA file, and in the definition of the network listener in the LISTENER.ORA file. Also known as *system ID*.

system ID

A synonym for instance identifier. Often abbreviated to SID.

SETLINKS

A utility to map symbolic links to logical partitions.

Startup

Operating System Dependent component that provides one-time configuration to startup functionality.

Also referred to as START.

START

An abbreviation for Startup.

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 database files. Every Oracle database has a tablespace called SYSTEM and may have additional tablespaces. A tablespace is used to group related logical structures together. For example, tablespaces commonly group all of an application's objects to simplify certain administrative operations.

TNSNAMES.ORA

A file that contains connect descriptors mapped to service names. The file may be maintained centrally or locally, for use by all or individual clients.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W