Oracle9*i*

Database Administrator's Guide

Release 2 (9.2) for Windows

March 2002

Part No. A95491-01

ORACLE®

Oracle9*i* Database Administrator's Guide, Release 2 (9.2) for Windows

Part No. A95491-01

Primary Author: Craig B. Foch

Contributing Authors: Herbert Kelly III, Mark Kennedy, and Helen Slattery

Contents

Send Us Your Comments ix

Preface xi

Audience xii
Organization xii
Related Documentation xiii
Conventions xiv
Documentation Accessibility xx

What's New in Oracle9*i* for Windows xxi

Oracle9*i* Release 2 (9.2) New Features xxi
Oracle9*i* Release 1 (9.0.1) New Features xxii

1 Postinstallation Database Creation

Before You Create a Database 1-2
Naming Conventions for Oracle Databases 1-2
Accessing Data Files and Log Files on Remote Computers 1-3
Creating a Database Using Database Configuration Assistant 1-3
Create a Database 1-4
Importing Sample Schemas 1-14
Delete a Database 1-15
Creating a Database Manually 1-15
Creating Directories 1-17
Exporting an Existing Database 1-17

Deleting Database Files 1-18
Modifying the Initialization Parameter File 1-19
Creating and Starting an Oracle Service 1-20
Putting the CREATE DATABASE Statement in a Script 1-21
Running the CREATE DATABASE Script 1-22
Importing a Database 1-23
Updating ORACLE_SID in the Registry 1-24
Backing Up the New Database 1-26
Using ORADIM Utility to Administer an Oracle Instance 1-27
Creating an Instance 1-29
Starting an Instance 1-30
Stopping an Instance 1-30
Modifying an Instance 1-31
Deleting an Instance 1-32

2 Postinstallation Configuration Tasks

Oracle *inter*Media 2-2
Configuring Oracle *inter*Media Audio, Video, Image, and Locator 2-3
Configuring Oracle *inter*Media Demos 2-4
Oracle Text 2-4
Configuring Oracle Text 2-5
Oracle Spatial 2-7
Manually Configuring Oracle Spatial 2-8
Shared Server Support 2-9
Enabling Shared Server Support for Java Clients 2-10
Enabling Shared Server Support for Two-Task Oracle Net Clients 2-10
Disabling Shared Server Support for Two-Task Oracle Net Clients 2-11
Advanced Replication 2-12
Checking Tablespace and Rollback Segment Requirements 2-13
Adding and Modifying Initialization Parameters 2-13
Monitoring Data Dictionary Tables 2-14

3 Administering a Database

Managing Oracle Services 3-2
Oracle Service Naming Conventions for Multiple Oracle Homes 3-2

Available Oracle Services 3-4
Starting Oracle Services 3-5
Stopping Oracle Services 3-6
Auto-starting Oracle Services 3-7
Starting and Shutting Down a Database with SQL*Plus 3-9
Starting and Shutting Down a Database Using Services 3-11
Running Multiple Instances 3-15
Creating and Populating Password Files 3-15
Viewing and Hiding the Password File 3-17
Connecting Remotely to the Database as SYS 3-19
Encrypting Database Passwords 3-19
Creating Control, Data, and Log Files on Remote Computers 3-20
Archiving Redo Log Files 3-20
Changing Archive Mode to ARCHIVELOG 3-21
Enabling Automatic Archiving 3-22
Using ORADEBUG Utility 3-24

4 Monitoring a Database

Database Monitoring Overview 4-2
Using Oracle for Windows NT Performance Monitor 4-2
Registry Information 4-3
Accessing Oracle for Windows NT Performance Monitor 4-4
Modifying Views 4-5
Understanding Oracle Performance Objects 4-6
Oracle for Windows NT Performance Monitor Troubleshooting Information 4-9
Using Event Viewer 4-9
Reading Event Viewer 4-11
Managing Event Viewer 4-12
Trace Files 4-13
Alert Files 4-13
Viewing Threads 4-14

5 Tuning Windows NT to Optimize Oracle9*i*

Windows NT Tuning Overview 5-2
Reduce Priority of Foreground Applications on Server Console 5-3

Configure Windows NT Server to Be an Application Server 5-4
Disable Unnecessary Services 5-6
Remove Unused Network Protocols 5-7
Reset Network Protocol Bind Order 5-8
Apply Latest Reliable Windows NT Server Service Pack 5-10
Use Hardware or Operating System Striping 5-11
Windows NT Server Stripe Set Example 5-12
Multiple Striped Volumes for Sequential and Random Access 5-16
Multiplex Windows NT Server Virtual Memory Paging File 5-16
Close All Unnecessary Foreground Applications 5-17
Startup Folder 5-17
Virtual DOS Machines 5-18
Screen Savers 5-18

6 Backing Up and Recovering Database Files

Selecting a Backup and Recovery Tool 6-2
Integrating RMAN with a Media Manager on Windows NT 6-4
Backing Up and Recovering Files with OCOPY Utility 6-5

7 Oracle9*i* Database Specifications for Windows

Initialization Parameter File Overview 7-2
Location of Initialization Parameter File 7-2
Editing Initialization Parameter File 7-3
Sample File 7-3
SGA_MAX_SIZE Parameter 7-4
Initialization Parameters Without Windows NT-Specific Values 7-5
Displaying Initialization Parameter Values 7-6
Uneditable Database Initialization Parameters 7-6
Calculating Database Limits 7-7

A Storing Tablespaces on Raw Partitions

Raw Partition Overview A-2
Physical Disk A-2
Logical Partition A-3

Physical Disk and Logical Partition Considerations A-4
Frequently Asked Questions..... A-4
Compatibility Issues..... A-5
Creating an Extended Partition A-5
Creating Logical Partitions in an Extended Partition..... A-7
Creating a Tablespace in a Raw Partition..... A-9

Glossary

Index

Send Us Your Comments

Oracle9*i* Database Administrator's Guide, Release 2 (9.2) for Windows

Part No. A95491-01

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

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

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

- Electronic mail: ntdoc_us@oracle.com
- FAX: (650) 506-7365 Attn: Oracle Database for Windows Documentation
- Postal service:
  Oracle Corporation
  Oracle Database for Windows Documentation Manager
  500 Oracle Parkway, Mailstop 1op6
  Redwood Shores, CA 94065
  USA

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

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

Preface

This guide is your primary source of introductory, postinstallation, configuration, and administration information for using Oracle9*i* Enterprise Edition, Oracle9*i* Standard Edition, or Oracle9*i* Personal Edition. Differences between the three versions are noted where appropriate.

This guide describes only the features of Oracle9*i* for Windows software that apply to the Windows NT, Windows 2000, Windows XP, and Windows 98 operating systems. Information on Oracle9*i* Personal Edition software on Windows 98 is not covered in this guide.

This preface contains these topics:

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

Audience

Oracle9i Database Administrator's Guide for Windows is necessary for anyone installing, configuring, or administering Oracle9*i* Enterprise Edition, Oracle9*i* Standard Edition or Oracle9*i* Personal Edition.

To use this document, you need:

- Windows NT or Windows 2000 installed and tested on your computer system
- Knowledge of object-relational database management concepts

Organization

This guide is organized as follows:

"What's New in Oracle9i for Windows"

Oracle9*i* release 2 (9.2) adds support for very large memory configurations and User Migration Utility, a new command-line tool. Oracle9*i* release 1 (9.0.1) added support for Windows XP Professional Edition, enhanced integration with Windows, and improvements in Database Configuration Assistant and Oracle Internet Directory administration. Server Manager and `CONNECT INTERNAL` were desupported in Oracle9*i* release 1 (9.0.1).

Chapter 1, "Postinstallation Database Creation"

This chapter describes how to create a database after installing Oracle, using either the Database Configuration Assistant or command-line tools.

Chapter 2, "Postinstallation Configuration Tasks"

This chapter describes some of the configuration tasks you must perform before using Oracle *inter*Media and other Oracle options.

Chapter 3, "Administering a Database"

This chapter describes how to administer Oracle9*i* for Windows.

Chapter 4, "Monitoring a Database"

This chapter describes how to monitor Oracle9*i* for Windows.

Chapter 5, "Tuning Windows NT to Optimize Oracle9i"

This chapter describes how to tune the Windows NT Server operating system to ensure that your Oracle9*i* database is running in the best possible environment.

Chapter 6, "Backing Up and Recovering Database Files"

This chapter provides information on backing up and recovering your database.

Chapter 7, "Oracle9i Database Specifications for Windows"

This chapter discusses initialization parameters, which Oracle9*i* uses on Windows NT to enable various features of the database every time an instance is started.

Appendix A, "Storing Tablespaces on Raw Partitions"

This appendix describes how to configure your system to store datafiles for tablespaces on raw partitions.

Glossary

Related Documentation

For more information, see these Oracle resources:

- *Oracle9i Database Installation Guide for Windows*
- *Oracle9i Database Release Notes for Windows*
- *Oracle9i Database Getting Started for Windows*
- *Oracle9i Database Administrator's Guide*
- *Oracle9i User-Managed Backup and Recovery Guide*
- *Oracle9i Database Migration*
- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle9i Net Services Administrator's Guide*
- *Oracle9i Database New Features*
- *Oracle9i Database Reference*
- *Oracle9i Database Error Messages*

In North America, printed documentation is available for sale in the Oracle Store at

```
http://oraclestore.oracle.com/
```

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

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

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

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://otn.oracle.com/admin/account/membership.html`

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://otn.oracle.com/docs/index.htm`

To access the database documentation search engine directly, please visit

`http://tahiti.oracle.com`

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.

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

Conventions in Code Examples

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

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

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

Convention	Meaning	Example
[ ]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (*digits* [, *precision*])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE | DISABLE}
|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]
...	Horizontal ellipsis points indicate either: ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code	CREATE TABLE ... AS *subquery*; SELECT *col1*, *col2*, ... , *coln* FROM employees;
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V$DATAFILE; NAME ---------- /fsl/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;

Convention	Meaning	Example

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

Conventions for Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program. For example, to start Database Configuration Assistant, you must click the Start button on the taskbar and then choose Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.	Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant
File and Directory Names	File and directory names are not case sensitive. The special characters <, >, :, ", /, |, and - are not allowed. The special character \ is treated as an element separator, even when it appears in quotes. If the file name begins with \ \, Windows assumes it uses the Universal Naming Convention.	`c:\winnt"\"system32` is the same as `C:\WINNT\SYSTEM32`
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is "^". Your prompt reflects the subdirectory in which you are working. Referred to as the command prompt in this guide.	`C:\oracle\oradata>`

Special characters	The backslash special character (\) is sometimes required as an escape character for the double quote (") special character at the Windows command prompt. Parentheses and the single quote special character (') do not require an escape character. See your Windows operating system documentation for more information on escape and special characters.	`C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"`
		`C:\>imp SYSTEM/`*`password`*
		`FROMUSER=scott TABLES=(emp, dept)`

HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`

Convention	Meaning	Example
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory that by default was: ■ `C:\orant` for Windows NT ■ `C:\orawin98` for Windows 98 or whatever you called your Oracle home. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is `C:\oracle\ora`*nn* where *nn* is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. See *Oracle9i Database Getting Started for Windows* for additional information on OFA compliance and for information on installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle Corporation does not own or control. Oracle Corporation neither evaluates nor makes any representations regarding the accessibility of these Web sites.

What's New in Oracle9*i* for Windows

This section describes new features of Oracle9*i* release 2 (9.2) and provides pointers to additional information. New features information from the previous release is also retained to help those users migrating to the current release.

The following sections describe new features:

- Oracle9i Release 2 (9.2) New Features
- Oracle9i Release 1 (9.0.1) New Features

Oracle9*i* Release 2 (9.2) New Features

This section contains these topics:

- Very Large Memory Support
- User Migration Utility

Very Large Memory Support

Oracle9*i* release 2 (9.2) for Windows supports Very Large Memory (VLM) configurations in Windows 2000 and Windows XP, which allows Oracle9*i* release 2 (9.2) to access more than the 4 gigabyte (GB) of RAM traditionally available to Windows applications. For more information, see "Oracle Scalability on Windows" in *Oracle9i Database Getting Started for Windows*.

User Migration Utility

A new command-line tool, User Migration Utility, simplifies conversion of local or external database users to enterprise users. For more information, see:

- "Database Tools Overview" in *Oracle9i Database Getting Started for Windows*
- "Manually Migrating Users" in *Oracle9i Security and Network Integration Guide*
- "Migrating Local or External Users to Enterprise Users" in *Oracle Advanced Security Administrator's Guide*

Oracle9*i* Release 1 (9.0.1) New Features

This section contains these topics:

- Windows XP Support
- Windows Integration
- Database Configuration Assistant Improvements
- Oracle Internet Directory Administration Improvements
- Using Oracle9i on Windows 2000
- CONNECT INTERNAL Not Supported
- Server Manager Not Supported

Windows XP Support

Oracle9*i* release 1 (9.0.1.1.1) for Windows is certified on the 32-bit version of Windows XP Professional Edition.

Oracle Corporation provides support information for components on various platforms, lists compatible client and database versions, and identifies patches and workaround information. Find the latest certification information at:

```
http://metalink.oracle.com/
```

You must register online before using Oracle*MetaLink*. After logging into Oracle*MetaLink*, select Product Lifecycle from the left-hand column.

Windows Integration

Oracle9*i* supports enhanced integration with Microsoft Transaction Services and Internet Information Services. Public key infrastructure and Single Sign-On capabilities in Oracle9*i* have also been integrated with Windows 2000, Active Directory, and Microsoft Certificate Store.

Oracle9*i* integration with Windows security supports Oracle Wallets in the registry and Active Directory, and it allows Oracle products to use Microsoft Certificate Store.

Synchronization between Active Directory and Oracle Internet Directory facilitates centralized scheduling and configuration of Oracle and third party meta-directory components.

Database Configuration Assistant Improvements

Database Configuration Assistant has been redesigned to include database definitions saved as templates. The templates can generate databases. Users can define new templates, modify existing templates, or use the ones Oracle provides. When creating a database with Database Configuration Assistant, users can include Oracle's new Sample Schemas.

Oracle Internet Directory Administration Improvements

Administration of Oracle Internet Directory replication server has been improved with addition of new replication queue management and reconciliation tools.

Using Oracle9*i* on Windows 2000

There are some differences between using Oracle9*i* on Windows 2000 and Windows NT 4.0. For more information, see "Using Oracle9*i* on Windows 2000" in *Oracle9i Database Getting Started for Windows*

CONNECT INTERNAL Not Supported

`CONNECT INTERNAL` and `CONNECT INTERNAL/PASSWORD` are not supported in Oracle9*i*. Use the following instead:

```
CONNECT / AS SYSDBA
CONNECT username/password AS SYSDBA
```

Server Manager Not Supported

Server Manager is not supported in Oracle9*i*. Use SQL*Plus instead. Most Server Manager scripts should work in a SQL*Plus environment, but some scripts may need to be modified.

1

Postinstallation Database Creation

This chapter describes how to create a database after installing Oracle, using either Database Configuration Assistant or command-line tools.

This chapter contains these topics:

- Before You Create a Database
- Creating a Database Using Database Configuration Assistant
- Creating a Database Manually
- Using ORADIM Utility to Administer an Oracle Instance

Before You Create a Database

Before you create a database, consider the following requirements:

- Naming Conventions for Oracle Databases
- Accessing Data Files and Log Files on Remote Computers

Naming Conventions for Oracle Databases

All mounted Oracle databases in a network must have unique database names. When a database is created, a name is associated with it and stored in its **control files**. If you provide the database keyword, either in the CREATE DATABASE statement or when prompted by Database Configuration Assistant, then that value becomes the name for that database.

If you attempt to **mount** two Oracle9*i* databases with the same database name, then you receive the following error during mounting of the second database:

```
ORA-01102: cannot mount database in EXCLUSIVE mode
```

If there are two or more Oracle9*i* databases on the same computer, but located in different Oracle homes, then the following rules apply:

- Each database name must be unique
- Each **SID** must be unique

To change the name of an existing database, you must use the CREATE CONTROLFILE statement to re-create your control files and specify a new database name. This restriction exists only for Oracle8*i* and later versions. Any Oracle7 instances running simultaneously with an Oracle9*i* **instance** are not subject to this restriction.

Note: Directory path examples in this chapter follow **Optimal Flexible Architecture (OFA)** guidelines. An example of an OFA path is *ORACLE_BASE**ORACLE_HOME*\rdbms\admin. If you specified non-OFA directories during installation, then your directory paths will differ.

Accessing Data Files and Log Files on Remote Computers

Oracle can access database files on a **remote computer** using Universal Naming Convention (UNC), but it may degrade database performance and network reliability. UNC is a PC format for specifying locations of resources on a local area network. UNC uses the following format:

```
\\server-name\shared-resource-path-name
```

For example, UNC specification for file `system01.dbf` in directory `C:\oracle\oradata\orcl` on shared server `argon` would be:

```
\\argon\oracle\oradata\orcl\system01.dbf
```

Locations of archive log files cannot be specified using UNC. Always set initialization parameter `LOG_ARCHIVE_DEST_`*n* to a mapped drive. If you set it to a UNC specification, then Oracle9*i* database does not start and you receive the following errors:

```
ORA-00256: error occurred in translating archive text string '\meldell\rmdrive'
ORA-09291: sksachk: invalid device specified for archive destination
OSD-04018: Unable to access the specified directory or device
O/S-Error: (OS 2) The system cannot find the file specified
```

An ORA-00256 error also occurs if you enter `\\\meldell\rmdrive` or `\\\meldell\\rmdrive`. Control files required the additional backslashes for Oracle8 release 8.0.4, but **redo log files** and datafiles did not.

Creating a Database Using Database Configuration Assistant

Oracle Corporation recommends you use Database Configuration Assistant to create a database, because it is easier.

Use Database Configuration Assistant (DBCA) to register a database running in a member server or workstation in Active Directory on a Windows 2000 domain from a member server or workstation.

If the database **service** is running on a server, then make sure `everyone` is a member of `Pre Windows 2000 Compatible` domain group. If `everyone` is not a member of this group, then add the **username**/computer name (LocalSystem) that the database service is running to the `Pre Windows 2000 Compatible` domain group. If this is not done, then the database on the member server will randomly get ACCESS DENIED errors when accessing Active Directory.

For Database Configuration Assistant to successfully register the database with Active Directory, the database service on the server is required to be running as a LocalSystem or domain user. If the database is running as a local user, then trying to register the database with Active Directory using Database Configuration Assistant fails, because this user cannot log on to Active Directory.

After successfully registering with the directory using Database Configuration Assistant, you must manually add a name to Access Control List of OracleDBSecurity container (in Active Directory) with read permissions on OracleDBSecurity container. If the database service is running as a LocalSystem, then manually add the computer name. If the database service is running as a domain user, then manually add the username. If this is not done, then you may not be able to use Active Directory to grant an enterprise **role**.

Database Configuration Assistant enables you to:

- Create a Database
- Configure Database Options in a database
- Delete a Database
- Manage Templates

Note: This chapter describes running Database Configuration Assistant in standalone mode (that is, after installation). It can also be used during installation to create a database.

Create a Database

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

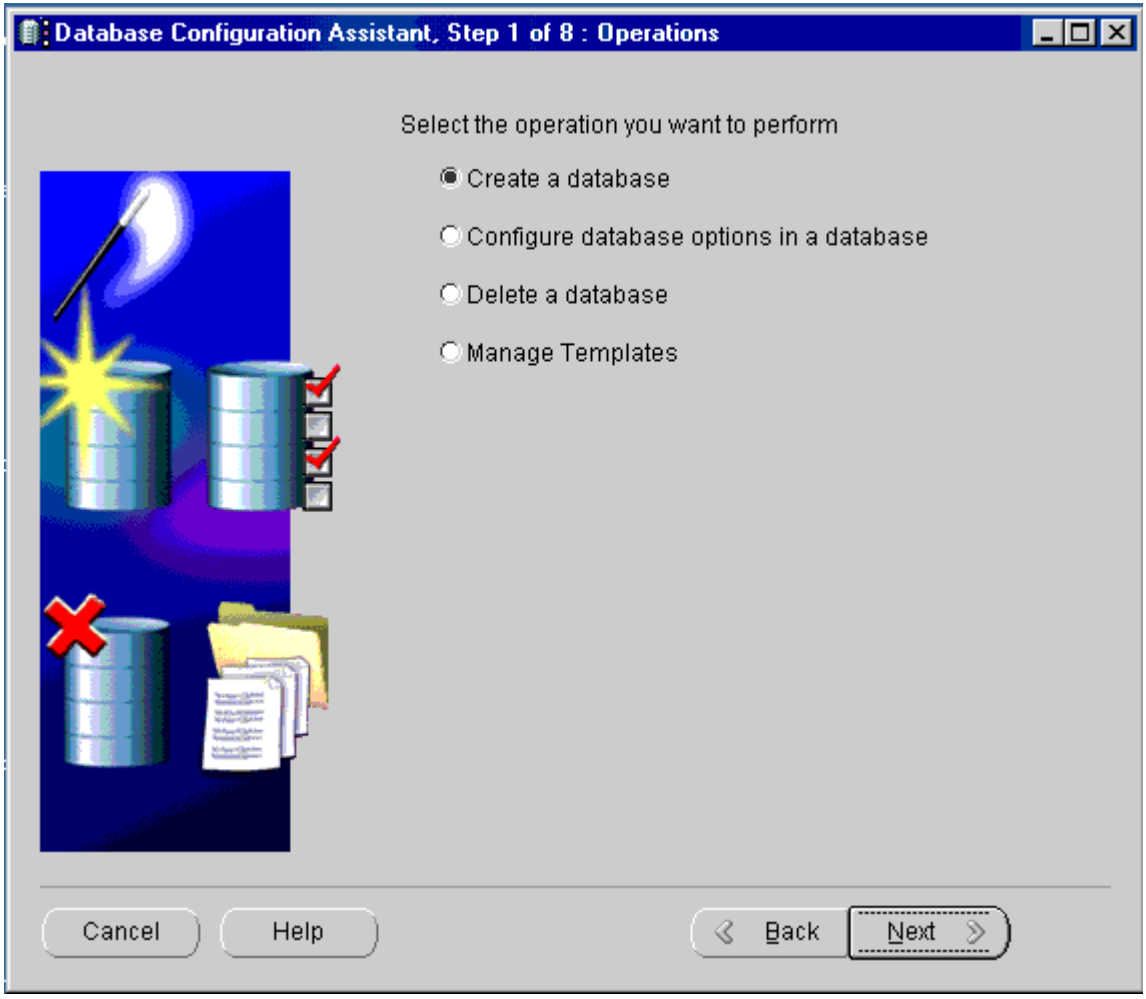
   The Database Configuration Assistant Welcome page appears.

Note: You must have the Windows Administrator **privilege** to create an Oracle9*i* database. If Database Configuration Assistant is run from a user account that is not part of the Administrator's group, then it displays a warning stating that you do not have administrative privileges to create the database.

---

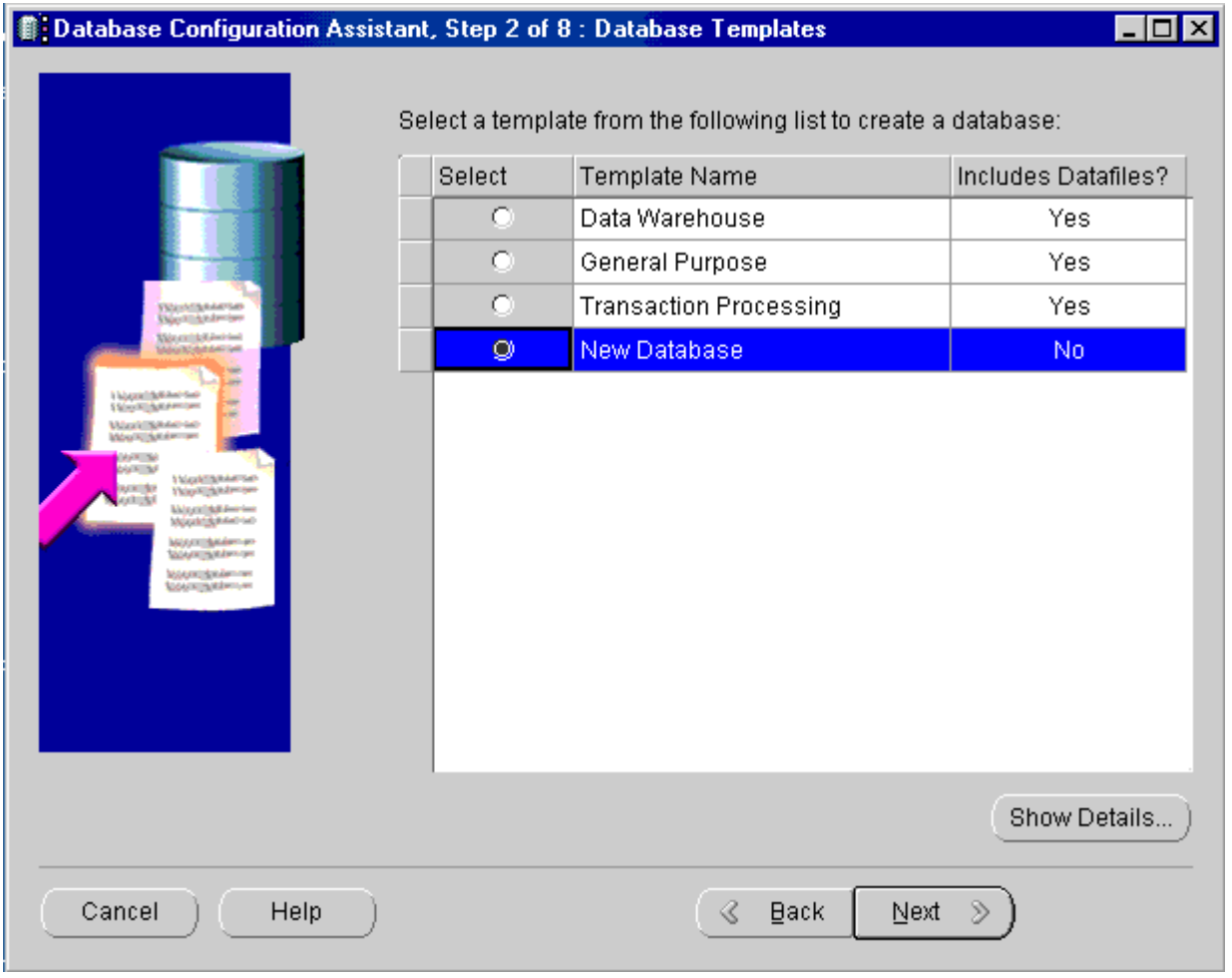
2. Click Next.

   A window appears prompting you to select an operation to perform.



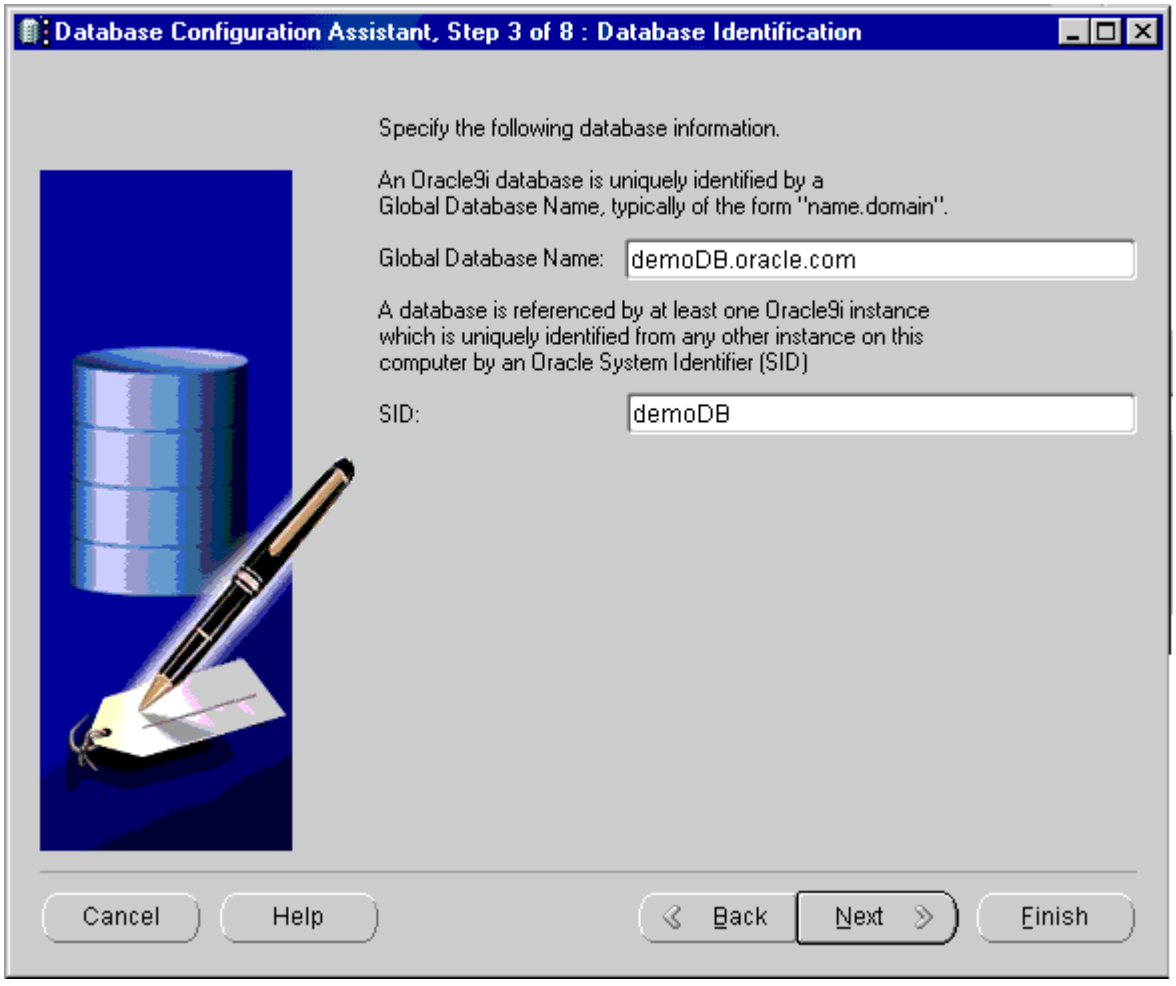
3. Select Create a database.
4. Click Next.

   A window appears prompting you to select a database template.



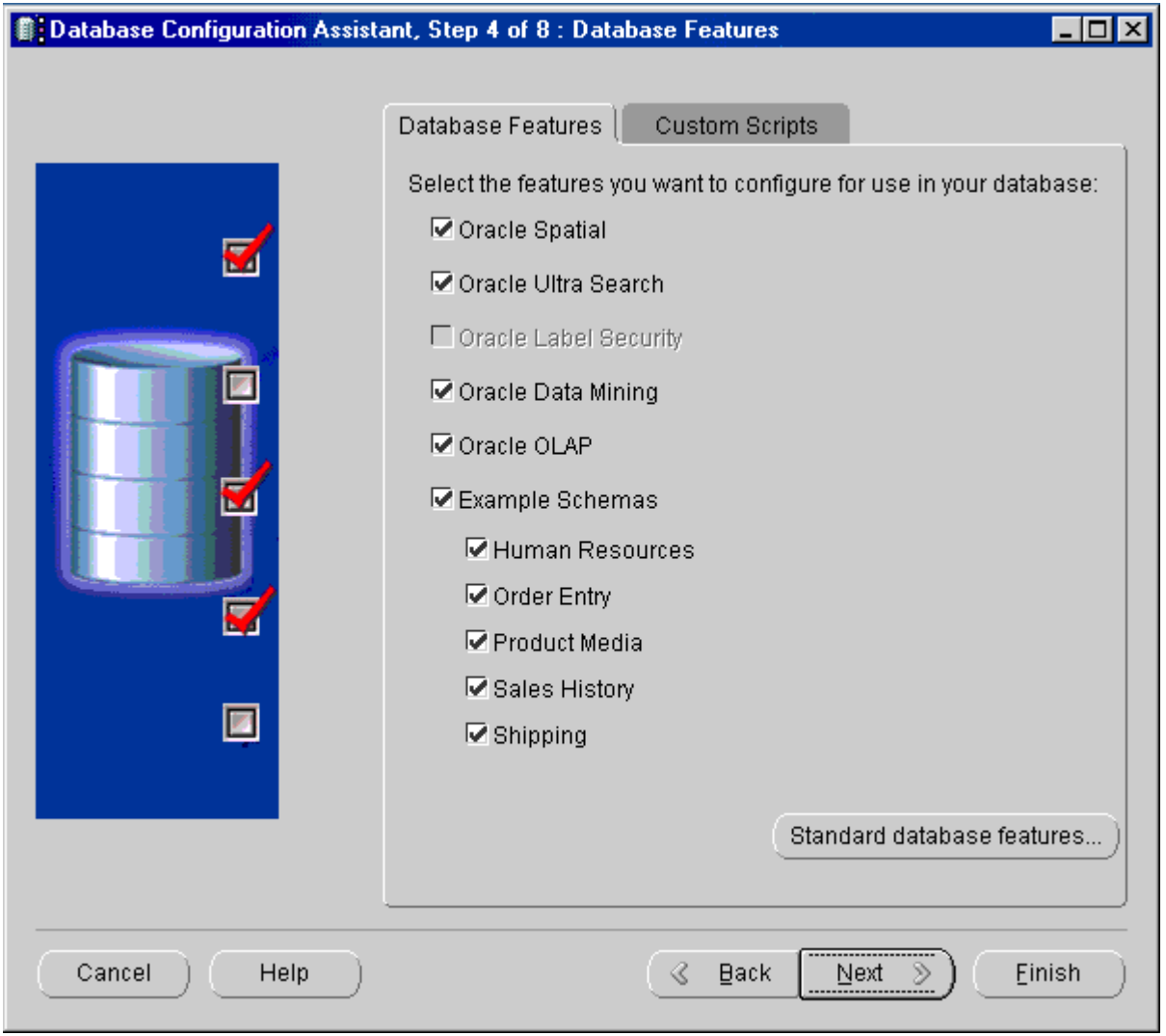
5. Select New Database.
6. Click Next.
7. Enter a global database name and an Oracle **System Identifier (SID)**.

   The name is typically of the form `name.domain`, and the SID must be 64 characters or less.



8. Click Next.

Note: If you use Database Configuration Assistant to create a new database in a new Oracle home, then file **listener.ora** is updated with the SID information, and a new TNS entry is generated in file **tnsnames.ora**. Both files are located in *ORACLE_BASE**ORACLE_HOME*\network\admin.



9. Your new database can have one or more of the following features:
   - Oracle Spatial
   - Oracle Ultra Search

- Oracle Data Mining
- Oracle OLAP
- Example **schema**

Note: Oracle Label Security is available only through the database Custom installation type. For more information, see *Oracle9i Database Installation Guide for Windows*.

Additional features become available if you click Standard Edition Features...

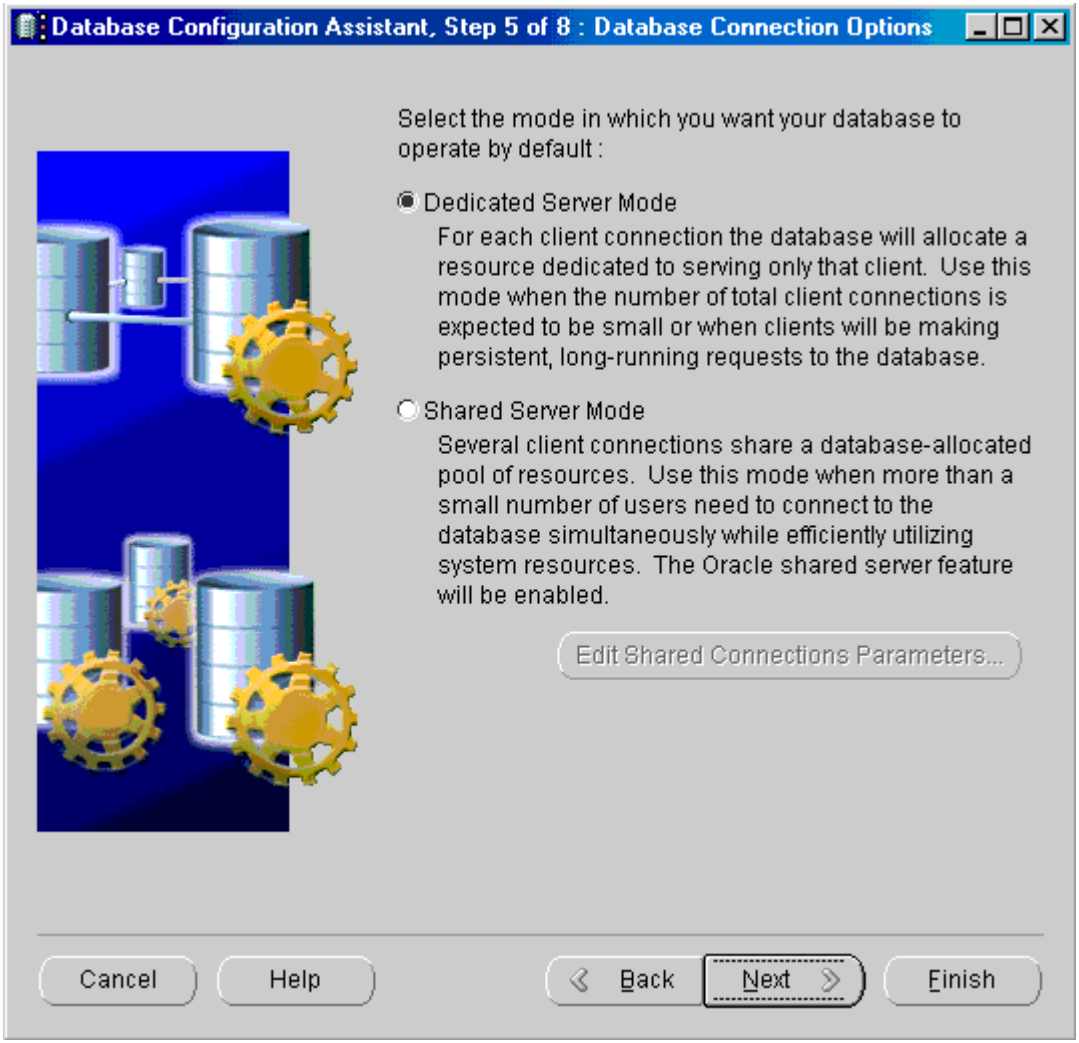
- Oracle JVM
- Oracle *inter*Media
- Oracle Text
- XDB Protocols

10. When you have selected the features to configure, click Next.

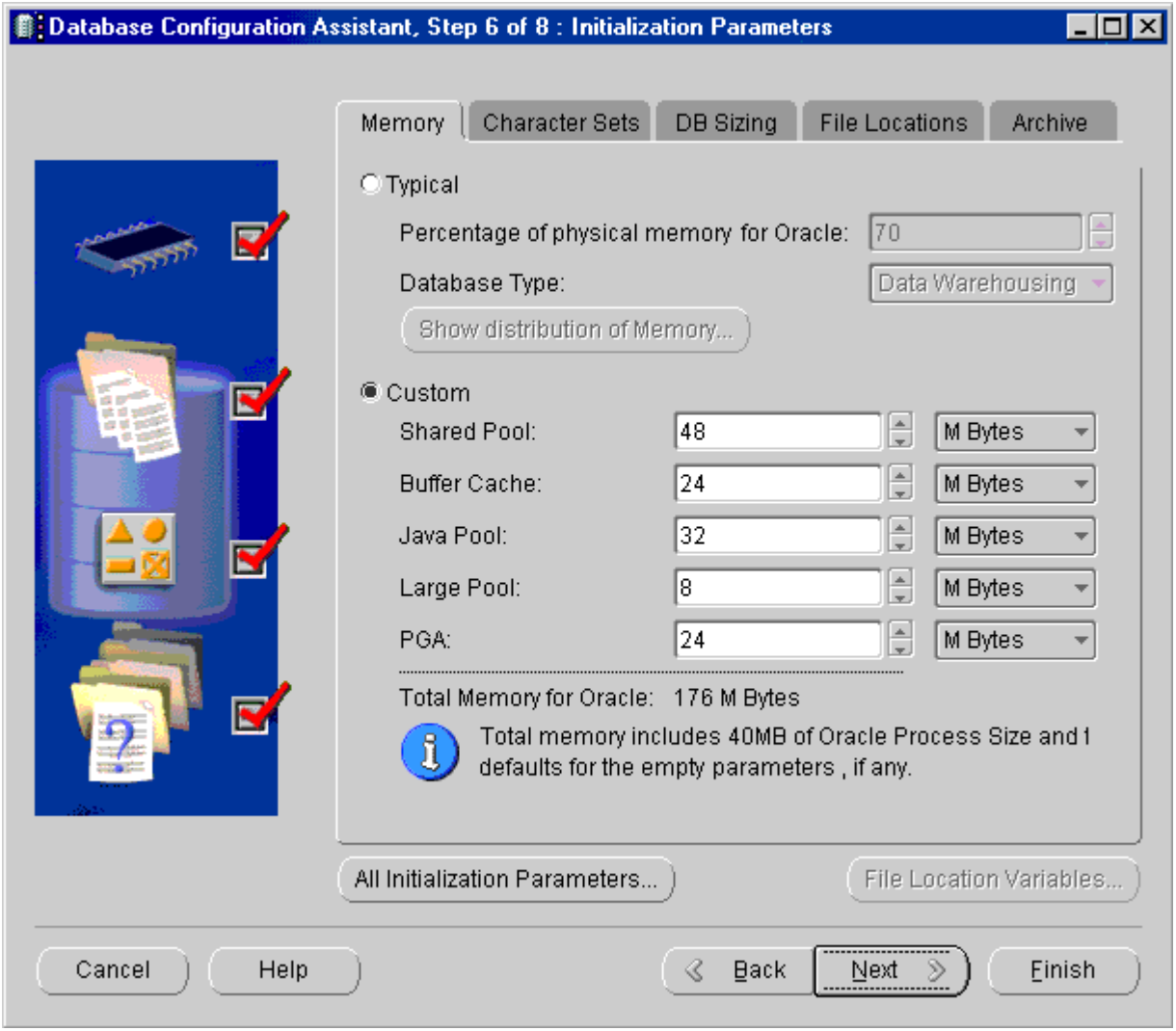
    The Database Connection Options window appears.

11. Select a database connection option:

    - In Dedicated Server Mode, each client connection to Oracle9*i* database allocates a resource dedicated to that client only.
    - In Shared Server Mode, several client connections share an allocated pool of resources.



12. Click Next.



13. Select an installation type.

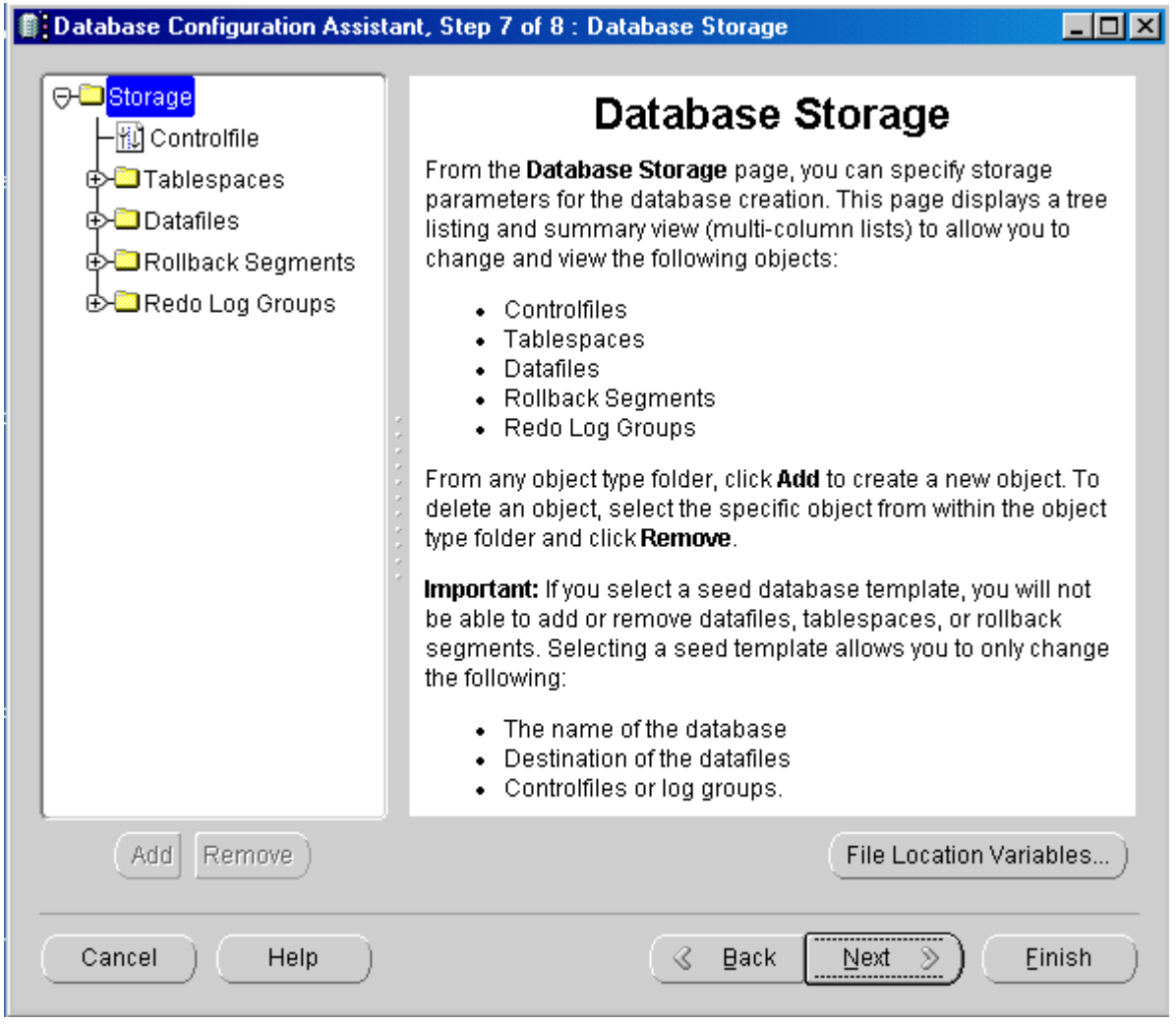
    Typical lets you set percentage of physical memory for Oracle and database type. Custom lets you specify initialization parameter values and other options including:

    - Shared Pool size

- Buffer Cache size
- Java Pool size
- Large Pool size

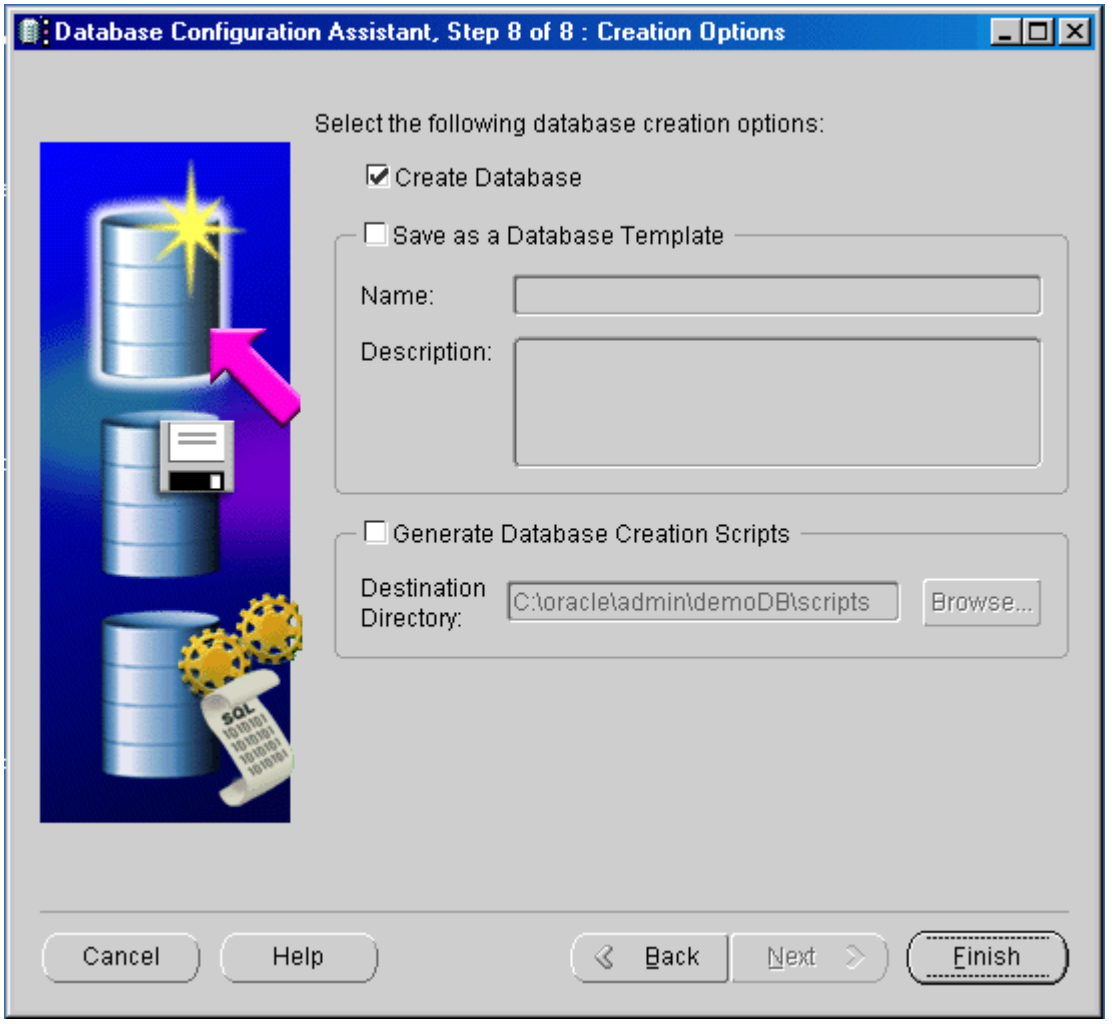
14. Click Next.

    The Database Storage page appears.



15. Follow the instructions on the page. When you finish, click Next.

The Creation Options window appears.



16. In this final step of Database Configuration Assistant, you select one or more creation options:
    - Create Database
    - Save as a Database Template
    - Generate Database Creation Scripts
17. Click Finish to start creation of Oracle9*i* database.

Importing Sample Schemas

After you finish creating your Oracle9*i* database with Database Configuration Assistant, you can import either or both of two sample schemas available on the CD-ROM:

- Online transaction processing (OLTP)
- Data warehousing

> **Note:** If you selected multipurpose, then you are not required to import sample schemas; they are already provided with your database.

To import an OLTP or data warehousing sample schema into your Oracle9*i* database:

1. Start SQL*Plus:

   ```
   C:\> sqlplus
   ```

2. Connect with the **SYSTEM** account:

   ```
   Enter user-name: SYSTEM/password
   ```

3. Create a special user account for importing the chosen schema. For the OLTP sample schema, enter:

   ```
   SQL> CREATE USER sampleoltp IDENTIFIED BY password;
   SQL> GRANT RESOURCE TO sampleoltp;
   SQL> GRANT CONNECT TO sampleoltp;
   ```

For the data warehousing sample schema, enter:

```
SQL> CREATE USER samplestar IDENTIFIED BY password;
SQL> GRANT RESOURCE TO samplestar;
SQL> GRANT CONNECT TO samplestar;
```

4. Exit SQL*Plus:

```
SQL> EXIT
```

5. Go to directory *ORACLE_BASE**ORACLE_HOME*\assistants\dbca\samples on your hard drive.

6. Import the chosen schema. For the OLTP sample schema, enter:

```
imp sampleoltp/password FILE=source90.dmp FULL=y LOG=myimp.log
```

For the data warehousing sample schema, enter:

```
imp samplestar/password FILE=target90.dmp FULL=y LOG=myimp.log
```

See Also: For more information on using Database Configuration Assistant:

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

Delete a Database

The Delete a Database option of Database Configuration Assistant lets you quickly and easily delete all database files except the **initialization parameter file**.

Creating a Database Manually

This section describes how to create a new database manually. As part of its database software files, Oracle provides a sample database creation script and a sample initialization parameter file, both of which can be edited to suit your needs. Alternatively, if you have an existing script you can use it as-is to create a database manually or edit it using the sample database creation script as a guide.

Database creations are of three types:

- Copy an existing database and delete the old database.
- Copy an existing database and keep the old database.
- Create a new database when no database exists on your system.

Table 1–1 summarizes tasks involved in creating a new database for each of these database creation categories. Each step is explained in detail in the following subsections.

Table 1–1 Manual Database Creation Tasks

Task	Copy existing database and delete old database	Copy existing database and keep old database	Create new database when no database exists on system
Creating Directories	Yes	Yes	Yes
Exporting an Existing Database	Yes	Maybe[1]	Not applicable
Deleting Database Files	Yes	No	Not applicable
Modifying the Initialization Parameter File	Yes	Yes	Yes
Creating and Starting an Oracle Service	No	Yes	Yes
Putting the CREATE DATABASE Statement in a Script	Yes	Yes	Yes
Running the CREATE DATABASE Script	Yes	Yes	Yes
Importing a Database	Yes	Maybe[2]	Not applicable
Updating ORACLE_SID in the Registry	No	Only if you change the default `SID`	Yes
Backing Up the New Database	Yes	Yes	Yes

[1] Yes if you copy data from the existing database to the new database; no otherwise.

[2] Yes if you import tables and other objects exported from the existing database; no otherwise.

We use an example in the following sections to demonstrate how to create a database. In this example, the existing database is the **starter database** with a `SID` of `orcl` located in directory `C:\oracle\oradata\orcl`. You will copy `orcl` to a new database with a database name and `SID` of `prod` located in directory `C:\oracle\oradata\prod`. You will then delete starter database `orcl`.

Note: In this example, *ORACLE_BASE* is `C:\oracle`. See "Conventions" on page xiv for more information on *ORACLE_BASE*.

Creating Directories

Create the following directories in which to put administration and database files for new database `prod`:

- `C:\oracle\admin\prod`
- `C:\oracle\admin\prod\bdump`
- `C:\oracle\admin\prod\pfile`
- `C:\oracle\admin\prod\udump`
- `C:\oracle\oradata\prod`

Exporting an Existing Database

You are required to export an existing database only if you intend to copy its contents to a new database. You will use Export utility for this task. Although you can start Export utility in either parameter mode or interactive mode, Oracle Corporation recommends parameter mode. Interactive mode provides less functionality than parameter mode and exists for backward compatibility only.

Example 1–1 Parameter Mode

```
C:\> exp SYSTEM/password FILE=myexp.dmp FULL=y LOG=myexp.log
```

Example 1–2 Interactive Mode

```
C:\> exp SYSTEM/password
```

Enter only the command `exp SYSTEM/password` to begin an interactive session and let Export utility prompt you for information it needs.

Note: If you use parameter mode, then Export utility considers filenames and directory names to be invalid if they contain one or more blank spaces. The workaround is to enclose the full path in the FILE= parameter in triple quotation marks. For example:

```
FILE="""C:\program files\export.dmp"""
```

If Export utility is used in interactive mode, then the filename or directory name can contain a space without quotation marks.

See Also: *Oracle9i Database Utilities* for more information on using Export utility.

To export all data from an existing database to a new database:

1. Set ORACLE_SID to the database service of the database whose contents you intend to export. For example, if the database you intend to export is starter database orcl, then enter the following at the command prompt. Note that there are no spaces around the equal sign (=) character.

   ```
   C:\> set ORACLE_SID=orcl
   ```

2. Start Export utility from the command prompt:

   ```
   C:\> exp SYSTEM/password FILE=myexp.dmp FULL=y LOG=myexp.log
   ```

   You now have a full database export of starter database orcl in file myexp.dmp. All messages from Export utility are logged in file myexp.log.

Deleting Database Files

Deleting database files is required only when you copy an existing database to a new database to replace the old database. In the following example, you delete the database files of starter database orcl.

To delete database files:

1. Shut down starter database orcl at the command prompt:

   ```
   C:\> oradim -SHUTDOWN -SID orcl -USRPWD password -SHUTTYPE inst -SHUTMODE i
   ```

2. Delete the following database files located in directory `C:\oracle\oradata\orcl`:

```
control01.ctl      example01.dbf      xdb01.dbf
control02.ctl      system01.dbf       redo01.log
control03.ctl      temp01.dbf         redo02.log
index01.dbf        tools01.dbf        redo03.log
drsys01.dbf        undotbs01.dbf
cwmlite01.dbf      user01.dbf
```

Modifying the Initialization Parameter File

If you are using starter database `orcl` as the basis for your new database, then copy the **initialization parameter file** from

```
C:\ORACLE_BASE\admin\orcl\pfile\init.ora
```

to

```
C:\ORACLE_BASE\admin\prod\pfile\init.ora
```

and modify the file as described in this section.

If you do not have an existing database on your system, then you cannot copy an existing initialization parameter file to use as the basis for your new initialization parameter file. However, you can use the sample initialization parameter file `initsmpl.ora` provided in

```
ORACLE_BASE\ORACLE_HOME\admin\sample\pfile
```

as the basis for the initialization parameter file for database `prod`.

If you use `initsmpl.ora` as the basis for the initialization parameter file, then you must set the following parameters to the indicated values, or you will not be able to start database `prod`:

- `DB_NAME=prod.domain`

  Parameter `DB_NAME` indicates the database name and must match the name used in the `CREATE DATABASE` statement in "Putting the CREATE DATABASE Statement in a Script" on page 1-21. You give a unique database name to each database. You can use up to eight characters for a database name. The name is not required to match the SID of the database service.

- `INSTANCE_NAME=prod.domain`
- `SERVICE_NAMES=prod.domain`
- `CONTROL_FILES = ("C:\oracle\oradata\prod\control01.ctl", "C:\oracle\oradata\prod\control02.ctl", "C:\oracle\oradata\prod\control03.ctl")`

  Parameter `CONTROL_FILES` lists database control files. You do not have control files on your file system at this point, because control files are created when you run the `CREATE DATABASE` statement. Ensure that you specify the complete path and filename, including drive letter.
- `BACKGROUND_DUMP_DEST = C:\oracle\admin\prod\bdump`
- `USER_DUMP_DEST = C:\oracle\admin\prod\udump`
- `DB_FILES=100`

  Modifying initialization parameter `DB_FILES` is not required, but it is recommended to optimize performance. Set this parameter to the same number as the value of the `MAXDATAFILES` option of the `CREATE DATABASE` statement. The value of `100` is used for this example.

  See Also: *Oracle9i Database Reference* for information on other initialization parameters that you can add or modify

Creating and Starting an Oracle Service

You are required to create and start an **Oracle service** only if you do one of the following:

- Copy an existing database to a new database and keep the old database
- Create a new database when you have no other database to copy

Before you create the database, first create a Windows service to run the database. This service is the Oracle9*i* database **process**, `oracle.exe`, installed in the form of a Windows NT service.

Use ORADIM utility to create the service. After it has been created, the service starts automatically. See "Using ORADIM Utility to Administer an Oracle Instance" on page 1-27 for information on how to use ORADIM utility.

To create and start an Oracle service:

1. Run ORADIM utility from the command prompt:

```
C:\> oradim -NEW -SID prod -INTPWD password -STARTMODE manual
-PFILE "C:\oracle\admin\prod\pfile\init.ora"
```

   Note that the previously created initialization parameter file is specified, with complete path, including drive name. You can check if the service is started in the Services window of the Control Panel.

2. Set ORACLE_SID to equal prod. Note that there are no spaces around the equal sign (=) character:

```
C:\> set ORACLE_SID=prod
```

Putting the CREATE DATABASE Statement in a Script

The CREATE DATABASE statement is a SQL statement that creates the database. A script containing this statement can be used anytime you create a database.

The CREATE DATABASE statement may have the following parameters:

- MAXDATAFILES - default value: 32, maximum value: 65534
- MAXLOGFILES - default value: 32, maximum value: 255

When you run a CREATE DATABASE statement, Oracle performs several operations depending upon clauses that you specified in the CREATE DATABASE statement or initialization parameters that you have set.

Note: Oracle Managed Files is a feature that works with the CREATE DATABASE statement to simplify administration of an Oracle database. Oracle Managed Files eliminates the requirement to directly manage operating system files comprising an Oracle database, because you specify operations in terms of database objects rather than filenames. For more information on using Oracle Managed Files see *Oracle9i Database Administrator's Guide*.

To create database prod, copy and save the following statement in a file named *script_name*.sql:

```
CREATE DATABASE prod
MAXLOGFILES 5
MAXDATAFILES 100
DATAFILE 'oracle\oradata\prod\system01.dbf' SIZE 325M REUSE
AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED
UNDO TABLESPACE "UNDOTBS" DATAFILE 'oracle\oradata\prod\undotbs01.dbf'
SIZE 200M REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED
CHARACTER SET WE8MSWIN1252
logfile 'C:\oracle\oradata\prod\redo01.log' size 100M reuse,
        'C:\oracle\oradata\prod\redo02.log' size 100M reuse,
        'C:\oracle\oradata\prod\redo03.log' size 100M reuse;
```

Running the CREATE DATABASE Script

To use the SQL script to create a database:

1. Verify that the service is started in the Control Panel. In this example, the service name is OracleServicePROD, and its status column must display Started. If not, then select the service name and choose Start.

   You can also check the status of the service by entering the following at the command prompt:

   ```
   C:\> net START
   ```

   A list of all Windows services currently running on the system appears. If OracleServicePROD is missing from the list, then enter:

   ```
   C:\> net START OracleServicePROD
   ```

2. Make PROD the current SID:

   ```
   C:\> set ORACLE_SID=PROD
   ```

3. Start SQL*Plus from the command prompt, and connect to the database as **SYSDBA**:

   ```
   C:\> sqlplus / NOLOG
   SQL> CONNECT / AS SYSDBA
   ```

   The message connected appears.

4. Turn on spooling to save messages:

   ```
   SQL> SPOOL script_name.log
   ```

5. Run script *script_name*.sql that you created in "Putting the CREATE DATABASE Statement in a Script" on page 1-21:

   ```
   SQL> @C:\oracle\ora92\rdbms\admin\script_name.sql;
   ```

   If the database is successfully created, then the instance is started and the following message appears numerous times: `Statement processed`

Importing a Database

You can use Import utility to import the full export created in "Exporting an Existing Database" on page 1-17 into the new database. Although you can start Import utility using either parameter mode or interactive mode, Oracle Corporation recommends parameter mode because it provides more functionality. Interactive mode exists solely for backward compatibility.

Example 1–3 Parameter Mode

```
C:\> imp SYSTEM/password FILE=myexp.dmp FULL=y LOG=myexp.log
```

Example 1–4 Interactive Mode

```
C:\> imp SYSTEM/password
```

Enter only `imp SYSTEM/password` to begin an interactive session and let Import utility prompt you for information it needs.

See *Oracle9i Database Utilities* for more information on using Import utility.

Note: If you use parameter mode, then Import utility considers filenames and directory names to be invalid if they contain one or more blank spaces. The workaround is to enclose the full path in the `FILE=` parameter in triple quotation marks. For example:

```
FILE="""C:\program files\export.dmp"""
```

If you use Import utility in interactive mode, then the filename or directory name can contain a space without quotation marks.

To import a database:

- Run Import utility:

```
C:\> imp SYSTEM/password FILE=myexp.dmp FULL=y LOG=myimp.log
```

> **IMPORTANT:** If the original database from which the export file was generated contains **tablespaces** that are not in the new database, then Import utility tries to create those tablespaces with associated datafiles.
>
> The easy solution is to ensure that both databases contain the same tablespaces. Datafiles are not required to be identical. Only tablespace names are important.

Updating ORACLE_SID in the Registry

If this is the first database on the system or if you intend to make the new database the default database, then you must make a change in the **registry**.

1. Start Registry Editor at the command prompt:

   ```
   C:\> regedt32
   ```

   The Registry Editor window appears.

2. Choose subkey `\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME0` for the first Oracle home on your computer. For subsequent installations to different Oracle homes on the same computer, the path is `\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME`*`ID`* where ID is the unique number identifying the Oracle home.

   > **See Also:** "Configuration Parameters and the Registry" in *Oracle9i Database Getting Started for Windows* for more information on subkey locations for **multiple Oracle homes**

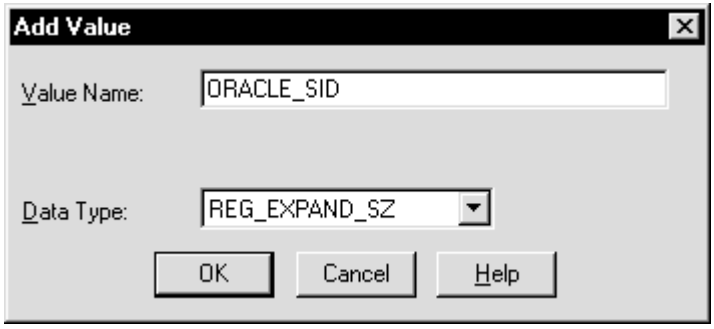
3. Locate parameter `ORACLE_SID` on the right side of the Registry Editor window.
4. Double-click the parameter name and change the data to the new SID, which is `prod` in this example.

If you do not yet have parameter ORACLE_SID, because this is the first database on your system, then you must create it.

To create parameter ORACLE_SID:

1. Choose Add Value from the Edit menu.

   The Add Value dialog box appears:



2. Enter ORACLE_SID in the Value Name text box.
3. Select REG_EXPAND_SZ (for an expandable string) in the Data Type list box.
4. Click OK.

   A string editor dialog box appropriate for the data type appears:

5. Enter prod in the String Editor dialog box.
6. Click OK.

   Registry Editor adds parameter ORACLE_SID.
7. Choose Exit from the Registry menu.

   Registry Editor exits.

Backing Up the New Database

> **Caution:** If anything goes wrong while operating the new database without a **backup**, then you must repeat the database creation procedure. Back up your database now to prevent loss of data.

To back up the new database:

1. Shut down the database instance and stop the service:

```
C:\> oradim -SHUTDOWN -SID prod -USRPWD password -SHUTTYPE srvc,inst -SHUTMODE i
```

> **Caution:** Although ORADIM utility returns the prompt immediately, you must wait for the database and the service to stop completely before continuing to Step 2. Wait until the Control Panel indicates service `OracleServicePROD` has stopped. If you do not do this, then the backup may be useless because it was taken while data was being written to datafiles.

2. Using the tool of your choice, back up database files.

   Database files consist of the initialization parameter file, control files, online redo log files, and datafiles.

   When the backup is complete, you can start the database again, create users and objects, if necessary, make any other changes, and use the database.

   Be sure to back up the database after making any significant changes, such as switching archiving mode or adding a tablespace or datafile.

   > **See Also:** Chapter 6, "Backing Up and Recovering Database Files", *Oracle9i Database Concepts*, *Oracle9i User-Managed Backup and Recovery Guide*, and *Oracle9i Database Administrator's Guide* for more information on archiving and backup and **recovery**.

> **Caution:** Do not store database files on a compressed drive. This can result in write errors and decreased performance.

Using ORADIM Utility to Administer an Oracle Instance

ORADIM utility is a command-line tool that is available with Oracle9*i* database. You are required to use ORADIM utility only if you are manually creating, deleting, or modifying databases. Database Configuration Assistant is an easier tool to use for this purpose.

The following sections describe ORADIM utility commands and parameters. Note that each command is preceded by a dash (-). To get a list of ORADIM utility parameters, enter:

```
oradim -? | -H | -HELP
```

Note: Specifying oradim without any options also returns a list of oradim parameters and descriptions.

When you use ORADIM utility, a log file called `oradim.log` opens in *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\database`, or in the directory specified by registry parameter `ORA_CWD`. All operations, whether successful or failed, are logged in this file. You must check this file to verify success of an operation.

If you have installed an Oracle database service on Windows 2000, then when logging in as SYSTEM user (LocalSystem), with startup mode set to Automatic, it is possible that the Oracle database service starts but the database does not start automatically. The following error message is written to file `ORADIM.LOG` in directory *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\database`:

```
ORA-12640: Authentication adapter initialization failed
```

Oracle Enterprise Management Agent, Oracle Enterprise Manager Management Server and Oracle Internet Directory may also fail, because they cannot connect to the database for the same reason.

The workarounds are to modify `SQLNET.ORA`, start the database with SQL*Plus after the service is started, or start the service as a specific user.

Modify SQLNET.ORA

Modify `SQLNET.ORA`, either by removing the line

```
sqlnet.authentication_services=(NTS)
```

or by changing it to

```
sqlnet.authentication_services=(NONE)
```

Start Database After Service Starts

Start the database manually after the Oracle database service has started, using SQL*Plus and connecting as SYSDBA.

Start Service as a Specific User

1. Choose Start > Settings > Control Panel > Services.

   The Services dialog box appears.

   > **Note:** If you are using Windows 2000, then you will not see Services in the Control Panel. Double-click Administrative Tools first, then double-click Services.

2. Select the service to start.
3. Click Startup.

   The Service Control dialog box appears.
4. On Windows NT:
   a. Select the service
   b. Click Startup
   c. Select This Account
5. On Windows 2000:
   a. Select the service
   b. Right-click and select Properties
   c. Select the Log On tab
   d. Select This Account
6. Specify the username and corresponding password.

Creating an Instance

To use ORADIM utility to create an instance, enter:

```
oradim -NEW -SID SID | -SRVC service_name [INTPWD password][-MAXUSERS
number][-STARTMODE auto | manual][-PFILE filename] [-TIMEOUT secs]
```

where

- `-NEW` indicates that you are creating a new instance. This is a mandatory parameter.
- `-SID` *`SID`* is the name of the instance to create. You must specify either this parameter or parameter `-SRVC`.
- `-SRVC` *`service_name`* is the name of the service to create (`OracleService`*`SID`*). You must specify either this parameter or parameter `-SID`.
- `-INTPWD` *`password`* is the password for the new instance. This is the password for the user logged in with SYSDBA privileges. Option `-INTPWD` is not required. If you do not specify it, then operating system authentication is used, and no password is required.
- `-MAXUSERS` *`number`* is the number of users defined in the password file. The default is 5.
- `-STARTMODE` `auto`, `manual` indicates whether to start the instance automatically or manually at startup. Default is `manual`.
- `-PFILE` *`filename`* is the initialization parameter file to be used with this instance. Ensure that you specify the complete path name of this file, including drive letter.
- `-TIMEOUT` *`secs`* sets the maximum time to wait (in seconds) before the service for a particular `SID` stops.

To create an instance called `PROD`, for example, you might enter:

```
C:\> oradim -NEW -SID prod -INTPWD mypassword1 -STARTMODE auto -PFILE
C:\oracle\admin\prod\pfile\init.ora
```

Starting an Instance

To use ORADIM utility to start an instance, enter

```
oradim -STARTUP -SID SID [-USRPWD user_pwd] [-STARTTYPE srvc | inst | srvc,
inst] [-PFILE filename]
```

where

- -STARTUP indicates that you are starting an instance that already exists. This is a mandatory parameter.
- -SID *SID* is the name of the instance to start. This is a mandatory parameter.
- -USERPWD *user_pwd* is the password.
- -STARTTYPE srvc, inst indicates whether to start the service or the instance. One or both values can be specified. If it is not specified, then the registry is checked for the current setting.
- -PFILE *filename* is the initialization parameter file to be used with this instance. Ensure that you specify the complete path name of this file, including drive letter.

To start an instance called puma, for example, you might enter:

```
C:\> oradim -STARTUP -SID puma -STARTTYPE inst -PFILE
C:\oracle\admin\prod\pfile\init.ora
```

Stopping an Instance

To use ORADIM utility to stop an instance, enter:

```
oradim -SHUTDOWN -SID SID [-USRPWD user_pwd] [-SHUTTYPE srvc | inst | srvc,
inst] [-SHUTMODE a | i | n]
```

where

- -SHUTDOWN indicates that you are stopping an instance. This is a mandatory parameter.
- -SID *SID* specifies the name of the instance to stop. This is a mandatory parameter.
- -USERPWD *user_pwd* specifies the password.

- -SHUTTYPE srvc, inst indicates whether to stop the service or the instance. One or both values can be specified. If it is not specified, then the registry is checked for the current setting.
- -SHUTMODE a, i, n specifies how to stop an instance: a indicates abort mode, i indicates immediate mode, and n indicates normal mode. This is an optional parameter. If you do not specify how to stop an instance, then normal is the default mode.

To stop an instance called puma, for example, you might enter:

```
C:\> oradim -SHUTDOWN -SID puma -SHUTTYPE srvc, inst
```

Modifying an Instance

You can modify an existing instance to change such values as instance name, instance password, startup mode, shutdown mode, and shutdown type. To use ORADIM utility to modify an instance, enter:

```
oradim -EDIT -SID SID [-NEWSID NEWSID][INTPWD password][-STARTMODE a | m][-PFILE
filename][SHUTMODE a | i | n][SHUTTYPE srvc | inst | srvc, inst]
```

where

- -EDIT indicates that you are modifying an instance. This is a mandatory parameter.
- -SID *SID* specifies the name of the instance to modify. This is a mandatory parameter.
- -NEWSID *NEWSID* specifies the new instance name. This is an optional parameter.
- -STARTMODE a, m specifies how to start the instance at system startup: a specifies automatically and m specifies manually. Default is manual.
- -PFILE *filename* specifies the initialization parameter file to be used with this instance. Ensure that you specify the complete path name of this file, including drive letter.

- -SHUTMODE a, i, n specifies how to stop an instance: a indicates abort mode, i indicates immediate mode, and n indicates normal mode. This is an optional parameter. If you do not specify how to stop an instance, then normal is the default mode.
- -SHUTTYPE srvc, inst indicates whether to stop the service or the instance. One or both values can be specified. If it is not specified, then the registry is checked for the current setting.

To change an instance name from prod to lynx, set a new instance password mycat123, and specify a new initialization parameter file, for example, you might enter:

```
C:\> oradim -EDIT -SID prod -NEWSID lynx -INTPWD mycat123 -STARTMODE a -PFILE
C:\oracle\admin\lynx\pfile\init.ora
```

Deleting an Instance

To use ORADIM utility to delete an instance, enter:

```
oradim -DELETE -SID SID | -SRVC service_name
```

where

- -DELETE Indicates that you are deleting an instance or service. This is a mandatory parameter.
- -SID *SID* specifies the name of the SID to delete. You must specify either this parameter or parameter -SRVC.
- -SRVC *service_name* specifies the name of the service to delete. You must specify either this parameter or parameter -SID.

To delete an instance called prod, for example, you might enter:

```
C:\> oradim -DELETE -SID prod
```

2

Postinstallation Configuration Tasks

This chapter describes some configuration tasks you must perform before using Oracle *inter*Media and other Oracle options. Where appropriate, the chapter provides references to other guides for those configuration tasks.

This chapter contains these topics:

- Oracle interMedia
- Oracle Text
- Oracle Spatial
- Shared Server Support
- Advanced Replication

Note: Directory path examples in this chapter follow **Optimal Flexible Architecture (OFA)** guidelines (for example, *ORACLE_BASE**ORACLE_HOME*\rdbms\admin). If you specified non-OFA compliant directories during installation, then your directory paths will differ. See "Multiple Oracle Homes and Optimal Flexible Architecture" in *Oracle9i Database Getting Started for Windows* for more information.

Oracle *inter*Media

Oracle *inter*Media includes the following components.

Client

Oracle *inter*Media Client provides an Oracle9*i* *inter*Media Audio, Image, and Video Java interface that lets you use local (client-side) applications to manipulate or modify multimedia data stored in a network-accessible (server-side) database. To use the Java library, set environment variable `CLASSPATH` so that it contains the Oracle *inter*Media library.

Client also provides a simple *inter*Media Image sample (`SimpImg.exe`) that was developed using Microsoft Visual C++. `SimpImg.exe` locates and updates images, using *inter*Media Image in an Oracle9*i* database.

Audio

Oracle *inter*Media Audio manages audio data in multiple file formats in an Oracle database. Types of audio data supported include conversations, songs, and other sounds in popular audio file formats. This makes it possible to integrate audio data with other application-specific object-relational data.

Video

Oracle *inter*Media Video manages video data in multiple video file formats. This makes it possible to integrate video data with other application-specific object-relational data.

Image

Oracle *inter*Media Image provides image storage, retrieval, and format conversion capabilities through an object data type (ODT). It also supports image storage, using Binary Large Objects (BLOBs), and references to image data residing in external files (BFILEs).

The Image component of Oracle *inter*Media also comes with a sample demonstration that shows how an image is extracted from an Oracle database.

Locator

Oracle *inter*Media Locator enables Oracle9*i* to support online internet-based geocoding facilities for locator applications and proximity queries.

Configuring Oracle *inter*Media Audio, Video, Image, and Locator

If you install Enterprise Edition, then Database Configuration Assistant starts automatically at the end of installation. If you choose any Database Configuration Assistant installation type other than Customized, then *inter*Media does not require manual configuration. All tasks described in this section are performed automatically.

If you select Customized installation, then Database Configuration Assistant will guide you through configuration of Oracle *inter*Media.

If you are creating and configuring a database manually, then you can configure Oracle *inter*Media Audio, Video, Image, and Locator as follows:

1. Start SQL*Plus:

   ```
   C:\> sqlplus / NOLOG
   ```

2. Connect to Oracle9*i* database with account SYS:

   ```
   SQL> CONNECT / AS SYSDBA
   ```

3. Start the database (if necessary):

   ```
   SQL> STARTUP
   ```

4. Run script `ordinst.sql`:

   ```
   SQL> @ORACLE_BASE\ORACLE_HOME\ord\admin\ordinst.sql
   ```

5. Run script `iminst.sql`:

   ```
   SQL> @ORACLE_BASE\ORACLE_HOME\ord\im\admin\iminst.sql
   ```

6. Exit SQL*Plus:

   ```
   SQL> EXIT
   ```

Note: If you manually copy your Oracle8*i* `listener.ora` and `tnsnames.ora` files into your Oracle9*i* network directory, then you must modify network configuration files `tnsnames.ora` and `listener.ora` on your server to enable **external routine** calls to work and *inter*Media to function properly. Follow the procedure in *Oracle9i Net Services Administrator's Guide*.

Configuring Oracle *inter*Media Demos

To configure Oracle *inter*Media Audio, Video, and Image demos, follow instructions in `readme.txt` files at locations shown in Table 2–1.

Table 2–1 interMedia Demo Instructions

Demo	File Location
Audio	*ORACLE_BASE**ORACLE_HOME*\ord\aud\demo
Video	*ORACLE_BASE**ORACLE_HOME*\ord\vid\demo
Image	*ORACLE_BASE**ORACLE_HOME*\ord\img\admin

To make *inter*Media Image demos for a Microsoft C compiler enter:

```
C:\> make
```

An additional demonstration resides in the following location:

*ORACLE_BASE**ORACLE_HOME*\ord\img\demo\vc\imgsamp\simpimg

To build and run the demonstration, you must first modify MAKEFILE to adapt it to your environment.

To configure Oracle *inter*Media Locator demonstrations, go to the following directory:

*ORACLE_BASE**ORACLE_HOME*\md\demo\geocoder

Sample data that can be loaded into Oracle9*i* is in `nh_cs.sql`. Examples that show use of Locator functionality are in `geohttp.sql` and `geolocat.sql`. Examples of data indexes created by using Locator are in `geoindex.sql`.

Oracle Text

Oracle Text enables text queries through SQL and **PL/SQL** from most Oracle interfaces. By installing Oracle Text with an Oracle database server, client tools such as SQL*Plus, Oracle Forms, and Pro*C / C++ are able to retrieve and manipulate text in an Oracle database.

Oracle Text manages textual data in concert with traditional data types in an Oracle database. When text is inserted, updated, or deleted, Oracle Text automatically manages the change.

Configuring Oracle Text

Your Oracle Text postinstallation tasks depend on your situation.

If you install Oracle Text from the CD-ROM and have a previous release of Oracle Text (formerly called *inter*Media Text) already installed, then see *Oracle Text Application Developer's Guide*.

If you **upgrade** your database, then you may be required to configure **Oracle Net** for external procedures. Otherwise, Oracle Text may not work. In any case other than upgrade, Oracle Net should be configured correctly by default to work with Oracle Text. See *Oracle Text Application Developer's Guide* and *Oracle Text Reference*.

If you are indexing formatted documents such as Microsoft Word, then you must set your environment to use the INSO filter before you can index your documentation set. For more information on setting up your environment for INSO filtering, see *Oracle Text Reference*.

Finally, if you install Oracle Text from the CD-ROM and do *not* have a previous release of Oracle Text installed, then Oracle9*i* database is already configured for use with Oracle Text if one of the following is true:

- The database is a **starter database** that you created by installing Enterprise Edition, Standard Edition or Personal Edition.
- The database is a starter database that you created by performing the following sequence of steps:

  1. Select Oracle9*i* Database in the Available Products window.
  2. Install Enterprise Edition, Standard Edition, or Personal Edition in the Installation Types window.
  3. Select General Purpose in the Database Configuration window.

     See Also: Chapter 1, "Postinstallation Database Creation" and *Oracle9i Database Installation Guide for Windows* for more information about creating a starter database

- You created the database by using Database Configuration Assistant in standalone mode, and selected Typical database creation type.

If none of these is true, then you must configure Oracle9*i* database for use with Oracle Text by doing one of the following:

- Using Database Configuration Assistant
- Configuring Manually

Using Database Configuration Assistant

To use Database Configuration Assistant to configure an Oracle9*i* database for use with Oracle Text at the time you create the database, simply select Oracle Text as the option to configure when prompted. To configure the database at a later time:

1. Start Database Configuration Assistant.

   Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

2. Select Configure database options in a database.
3. Select the database to modify when prompted.
4. Select Oracle Text as the option to configure when prompted.

Configuring Manually

Manually configuring an Oracle database for use with Oracle Text consists of creating a tablespace for Oracle Text **data dictionary** tables and then creating **username** `ctxsys` and Oracle Text data dictionary tables.

1. Start SQL*Plus:

   ```
   C:\> sqlplus / NOLOG
   ```

2. Connect Oracle9*i* database with account SYSDBA:

   ```
   SQL: CONNECT / AS SYSDBA
   ```

3. Create a tablespace for Oracle Text data dictionary tables:

   ```
   SQL> CREATE TABLESPACE tablespace_name DATAFILE 'ORACLE_BASE\oradata\db_
   name\drsys01.dbf' SIZE 80m;
   ```

4. Connect AS SYSDBA:

   ```
   SQL> CONNECT USERNAME/PASSWORD AS SYSDBA
   ```

5. Run script `dr0csys.sql` to create username `ctxsys`:

   ```
   SQL> @ORACLE_BASE\ORACLE_HOME\ctx\admin\dr0csys.sql password default_
   tablespace_name temporary_tablespace_name;
   ```

   where:

   - *password* is the password that you intend to use for username `ctxsys`.
   - *default_tablespace_name* is the default tablespace for Oracle Text data dictionary tables. Set the default tablespace to the value of *tablespace_name* in step 3.
   - *temporary_tablespace_name* is the temporary tablespace for Oracle Text data dictionary tables. Set the temporary tablespace to the value of *tablespace_name* in step 3.

6. Connect as `ctxsys`:

   ```
   SQL> CONNECT ctxsys/password
   ```

7. Run script `dr0inst.sql` to create and populate Oracle Text data dictionary tables:

   ```
   SQL> @ORACLE_BASE\ORACLE_HOME\bin\oractxx9.dll;
   ```

8. Run the language-specific default script, where *xx* is the language code (for example, `us`):

   ```
   SQL> @ORACLE_BASE\ORACLE_HOME\ctx\admin\defaults\drdefxx.sql;
   ```

9. Exit SQL*Plus:

   ```
   SQL> EXIT
   ```

Oracle Spatial

Oracle Spatial makes storage, retrieval, and manipulation of spatial data easier and more intuitive to users.

One example of spatial data is a road map. A road map is a two-dimensional object that contains points, lines, and polygons representing cities, roads, and political boundaries such as states. A road map represents geographic information. Locations of cities, roads, and political boundaries are projected onto a two-dimensional display or piece of paper, preserving relative positions and relative distances of objects.

If you install Oracle Spatial through Enterprise Edition, then no manual configuration is required. All Oracle Spatial configuration tasks are performed automatically.

If you install both Oracle Spatial and Oracle9*i* database together through Enterprise Edition or Standard Edition Custom installation, then Database Configuration Assistant starts automatically at the end of installation. If you choose Custom installation and select Create new database, then the assistant asks if Oracle Spatial is to be configured automatically.

If you install Oracle Spatial during a separate installation from Enterprise Edition, then you must either start Database Configuration Assistant and select Configure database options in a database or configure Oracle Spatial manually.

Manually Configuring Oracle Spatial

1. Start SQL*Plus at the command prompt:

   ```
   C:\> sqlplus / NOLOG
   ```

2. Connect to Oracle9*i* database with account SYSDBA:

   ```
   SQL> CONNECT / AS SYSDBA
   ```

3. Start the database (if necessary):

   ```
   SQL> STARTUP
   ```

4. Run script `ordinst.sql`:

   ```
   SQL> @ORACLE_BASE\ORACLE_HOME\ord\admin\ordinst.sql
   ```

5. Connect to the database as user **SYSTEM**:

   ```
   SQL> CONNECT SYSTEM/password
   ```

6. Run script `mdinst.sql`:

   ```
   SQL> @ORACLE_BASE\ORACLE_HOME\md\admin\mdinst.sql
   ```

7. Exit SQL*Plus:

   ```
   SQL> EXIT
   ```

Note: Script `mdinst.sql` has a variable `%MD_SYS_PASSWORD%` that Oracle Universal Installer will **instantiate** at installation time. Therefore, if you have changed the `mdsys` user's password, then be sure during a manual installation to remember also to update script `mdinst.sql` with that password.

Shared Server Support

Database Configuration Assistant lets you enable or disable shared server support in your Oracle9*i* database. If your Oracle9*i* database is not configured for shared server mode, then it is configured for dedicated server mode.

When Oracle9*i* database is in dedicated server mode, each client connection is allocated a resource dedicated to serving that client. Dedicated server mode is recommended if:

- The database is used for data warehousing
- A small number of users will connect to the database

Shared server mode (formerly called multithreaded server mode) enables many client user processes to share a small number of server processes. Many client users can connect to a dispatcher **process**. The dispatcher process then routes client requests to the next available shared server process. No dedicated server process exists for each client user process for the duration of the connection. Instead, inactive server processes are recycled and used as needed. This reduces system overhead and enables you to increase the number of supported users.

Shared server mode is recommended if

- The database is used for online transaction processing (OLTP)
- A large number of users will simultaneously connect to the database
- You intend to use Oracle Net features such as connection pooling, connection multiplexing, and load balancing
- Careful management and use of system resources is important
- Predictable and fast database connection times are important (as in Web applications)

Note: Shared server mode is required for proper functioning of Oracle9*i* Java Virtual Machine.

By your choice of install options and use of Database Configuration Assistant, you can set shared server support for Java clients, two-task Oracle Net clients, or both.

If you install a database through Enterprise Edition, then the database is automatically configured for shared server mode for Java clients and dedicated server mode for two-task Oracle Net clients.

If you install a database through the typical option of Database Configuration Assistant and select Oracle9*i* JVM, then the database is configured for shared server mode for Java clients and dedicated server mode for two-task Oracle Net clients.

If you install a database through the Custom option of Database Configuration Assistant, then the database is configured for dedicated server mode or shared server mode, depending on what you select when prompted by the assistant.

Enabling Shared Server Support for Java Clients

Shared server support for Java clients is automatically enabled when you install Oracle9*i* JVM.

Enabling Shared Server Support for Two-Task Oracle Net Clients

To create a new database and enable shared server mode for two-task Oracle Net clients:

1. Start Database Configuration Assistant.

   Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

   The Database Configuration Assistant Welcome page appears.
2. Select Create a database.
3. Select New database.
4. Select Shared Server Mode. Click Edit Shared Connections Parameters.
5. On the Basic tab of the Shared Server Mode page enter 20 or more for maximum number of server processes.

To enable shared server support for two-task Oracle Net clients on an existing database:

1. Start Database Configuration Assistant.

   Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

The Database Configuration Assistant Welcome page appears.

2. Click Next.

   A window appears prompting you to select an operation to perform.

3. Select Configure database options in a database. Click Next.
4. Select the Oracle9*i* database to modify and, if prompted, enter the SYS password.
5. Click Next.
6. Select Shared Server Mode and click Edit Shared Connections Parameters.
7. On the Basic tab of the Shared Server Mode page make any necessary changes to shared server parameters. Click Help for more information about the parameters.
8. Click OK.
9. Click Finish.

   A dialog box prompts you to select the **initialization parameter file** to use.

10. Select the appropriate file and choose OK.

    Your initialization parameter file is modified.

11. Shut down and restart your Oracle9*i* database for the changes to take effect.

Disabling Shared Server Support for Two-Task Oracle Net Clients

Shared server support for two-task Oracle Net clients must be disabled manually.

To disable shared server support for two-task Oracle Net clients:

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

   The Database Configuration Assistant Welcome page appears.

2. Select Configure database options in a database and choose Next.
3. Select the Oracle9*i* database to modify.
4. Click Next.
5. Select Dedicated Server Mode and choose Finish.

   A dialog box prompts you to select the initialization parameter file to use.

6. Select the appropriate file and choose OK.

   Your initialization parameter file is modified.

7. Shut down and restart your Oracle9*i* database for the changes to take effect.

Advanced Replication

Oracle9*i* installs **replication** packages and procedures automatically rather than as a separate manual process. This section describes how to manually configure Advanced Replication in your Oracle9*i* database.

Follow the instructions *only* if you add Advanced Replication to an Oracle9*i* database that was not previously configured with this feature.

Configuring Advanced Replication consists of the following steps:

- Checking Tablespace and Rollback Segment Requirements
- Adding and Modifying Initialization Parameters
- Monitoring Data Dictionary Tables

> **See Also:** There are many configuration and usage possibilities with Advanced Replication. For more information about Advanced Replication and for definitions of master sites and materialized view sites, see the following guides:
>
> - *Oracle9i Replication*
> - *Oracle9i Database Concepts*
> - *Oracle9i Database Administrator's Guide*

Checking Tablespace and Rollback Segment Requirements

Recommended tablespace and rollback segment requirements for Advanced Replication are shown in Table 2–2.

Table 2–2 Advanced Replication Tablespace/Rollback Segment Requirements

Tablespace/Rollback Segment	Minimum Free Space
SYSTEM[1]	20 MB
UNDO TABLESPACE	10 MB
RBS	5 MB
TEMP	10 MB
USERS	No specific requirement

[1] Replication triggers and procedures are stored here.

Adding and Modifying Initialization Parameters

If you use Advanced Replication, then certain parameter values must be added to the initialization parameter file, and others must be set to recommended values. Parameter names and values for the master site and materialized view sites are shown in Table 2–3.

Table 2–3 Advanced Replication Initialization Parameters

Parameter Name	Recommended Value	Site
JAVA_POOL_SIZE	50 MB	master
DISTRIBUTED_LOCK_TIMEOUT	300 seconds	master
GLOBAL_NAMES	TRUE	master
OPEN_LINKS	4	master
PROCESSES	Add 9 to current value	master
JOB_QUEUE-PROCESSES	2[1]	master
JOB_QUEUE_PROCESSES	2	materialized view

[1] Depends on number of n-way sites.

Monitoring Data Dictionary Tables

If you use Advanced Replication and intend to set up a large number of replicated objects, then you are required to monitor the following data dictionary tables with the SQL SELECT command:

- ARGUMENT$
- IDL_CHAR$
- IDL_UB1$
- IDL_UB2$
- IDL_SB4$
- I_ARGUMENT1
- I_SOURCE1I$
- SOURCE$
- TRIGGER

If necessary, increase storage parameters to accommodate storage requirements of large numbers of replicated objects.

3

Administering a Database

This chapter describes how to administer Oracle9*i* for Windows.

This chapter contains these topics:

- Managing Oracle Services
- Starting and Shutting Down a Database with SQL*Plus
- Starting and Shutting Down a Database Using Services
- Running Multiple Instances
- Creating and Populating Password Files
- Connecting Remotely to the Database as SYS
- Encrypting Database Passwords
- Creating Control, Data, and Log Files on Remote Computers
- Archiving Redo Log Files
- Using ORADEBUG Utility

Managing Oracle Services

This section provides information on the following:

- Oracle Service Naming Conventions for Multiple Oracle Homes
- Available Oracle Services
- Starting Oracle Services
- Stopping Oracle Services
- Auto-starting Oracle Services

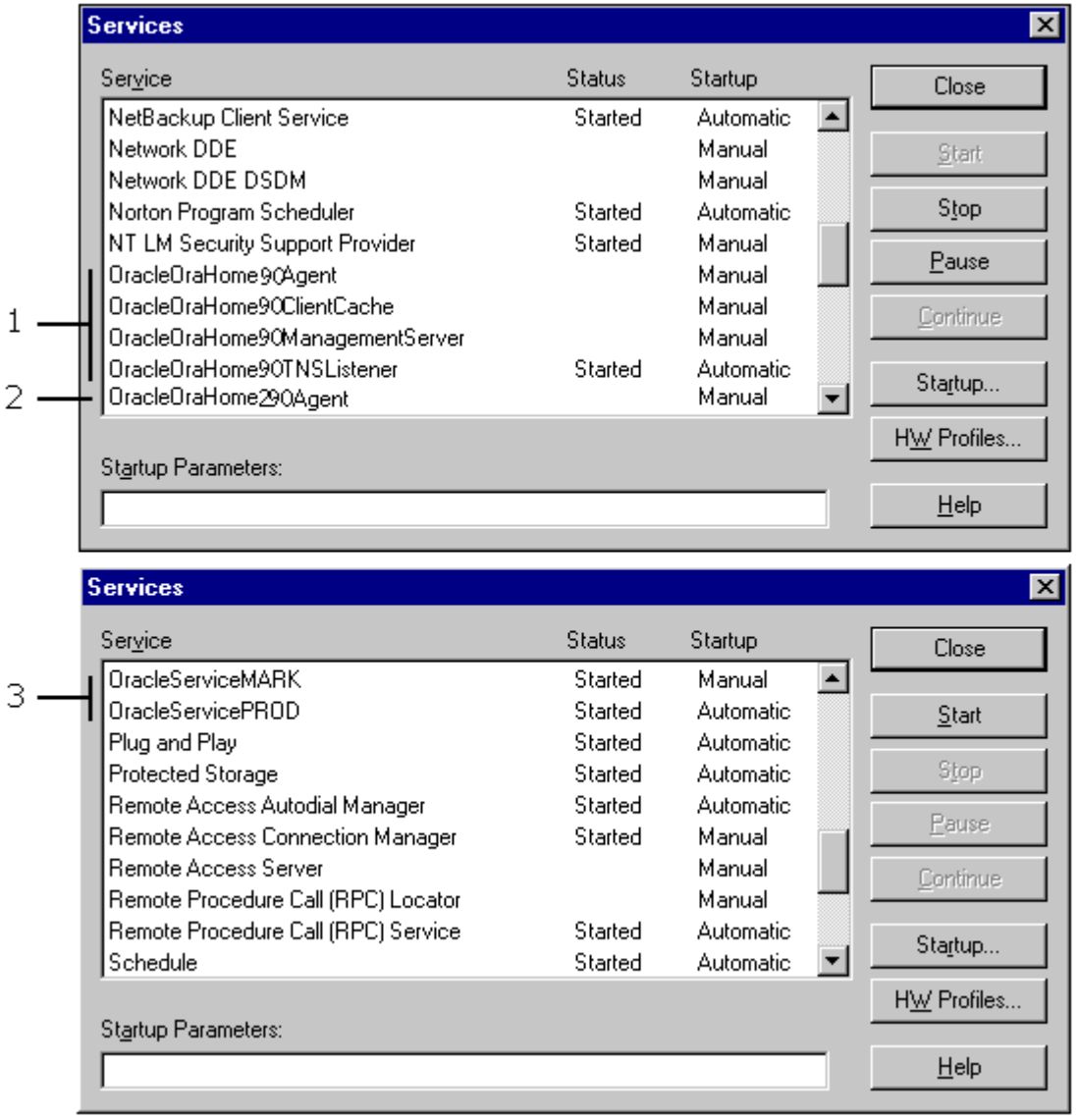
Oracle Service Naming Conventions for Multiple Oracle Homes

Oracle9*i* for Windows lets you have **multiple Oracle homes** on a single computer. This feature, described in "Database Tools Overview" in *Oracle9i Database Getting Started for Windows*, affects **Oracle service** naming conventions. As you perform installations into Oracle home directories:

- You must accept default Oracle home name provided or specify a different name for each Oracle home directory.
- You are prompted to give a **System Identifier (SID)** and global database name for each database installation.

Figure 3–1 shows how the Services dialog box appears with two Oracle9*i* databases on a single computer. A home name appears in most service names. Line 1 points to service names for the first Oracle9*i* database installation (OraHome90). Line 2 points to service names for the second Oracle9*i* database installation (OraHome290). Line 3 points to services with SIDs of MARK and PROD.

Figure 3–1 Services Dialog Boxes



Available Oracle Services

Depending on products that you have installed, a number of Oracle services are started when you restart your Windows computer. A user with a non-system account must have local administrative rights to run services on a Windows computer. The five main Oracle services are described in this section. For information on other available services, see "Oracle9*i* Services on Windows" in *Oracle9i Database Getting Started for Windows*.

OracleService*SID*

Created for the database instance System Identifier (SID), where *SID* is the value you entered for the database name during Oracle9*i* installation. This service is mandatory. If is it not started, then the following `ORA-12560` error message appears when you attempt to use any of Oracle9*i* Utilities, such as SQL*Plus:

```
ORA-12560 TNS: protocol adapter error
```

Oracle*HOME_NAME*TNSListener

Listens for and accepts incoming connection requests from client applications. Automatically starts when the Windows computer restarts. If it is not started, then the following error message appears when attempting to connect with `username/password@net_service_name`:

```
ORA-12541 TNS: no listener
```

Oracle*HOME_NAME*Agent

Part of the Oracle Enterprise Manager product suite. Intelligent Agent is required for execution of jobs and performance monitoring of Oracle services or targets such as listeners, databases, Oracle HTTP Server, and Oracle Applications. Intelligent Agent also collects statistical data for Capacity Planner and Performance Manager, which are data-collecting applications available in Oracle Diagnostics Pack.

Oracle*HOME_NAME*HTTPServer

Starts Oracle HTTP Server. Used as the Web server with browser-based Oracle Enterprise Manager.

Oracle*HOME_NAME*ManagementServer

Management Server is the middle tier of Oracle Enterprise Manager. It maintains centralized intelligence and distributed control between clients and managed nodes.

Starting Oracle Services

Oracle services must be started for you to use Oracle9*i* database and its products. You can start Oracle services from three different locations:

- Control Panel
- Command Prompt
- Oracle Administration Assistant for Windows NT

> **Note:** You can start Oracle9*i* database when you start `OracleService`*`SID`*. See "Starting and Shutting Down a Database Using Services" on page 3-11 for information on **registry** parameters that enable you to do this.

Control Panel

To start Oracle services from the Control Panel:

1. Choose Start > Settings > Control Panel.

   The Control Panel window appears.

2. Double-click Services.

   The Services dialog box appears.

3. Find the service to start in the list, select it, and choose Start.

   If you cannot find `OracleServiceSID` in the list, then use ORADIM utility to create it.

4. Click Close to exit the Services dialog box.

> **Note:** If you are using Windows 2000, then you will not see Services in the Control Panel. Double-click Administrative Tools first, then double-click Services.

Command Prompt

To start Oracle services from the command prompt, enter:

```
C:\> NET START service
```

where *`service`* is a specific service name, such as `OracleServiceORCL`.

Oracle Administration Assistant for Windows NT

To start Oracle services from Oracle Administration Assistant for Windows NT:

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Administration Assistant for Windows NT.
2. Right-click the *SID*.

   where *SID* is a specific instance name, such as `orcl`.
3. Click Start Service

   This starts service `OracleServiceORCL`.

Stopping Oracle Services

On occasion (for example, when re-installing Oracle9*i* database), you must stop Oracle services. You can stop Oracle services from three different locations:

- Control Panel
- Command Prompt
- Oracle Administration Assistant for Windows NT

> **Note:** You can stop Oracle9*i* database in normal, immediate, or abort mode when you stop `OracleService`*SID*. See "Starting and Shutting Down a Database Using Services" on page 3-11 for information on registry parameters that enable you to do this.

Control Panel

To stop Oracle services from the Control Panel:

1. Choose Start > Settings > Control Panel.

   The Control Panel window appears.
2. Double-click Services.

   The Services dialog box appears.
3. Select `Oracle`*HOME_NAME*`TNSListener` and choose Stop.

   `Oracle`*HOME_NAME*`TNSListener` is stopped.
4. Select `OracleService`*SID* and choose Stop.

5. Click OK.

   `OracleService`*`SID`* is stopped.

Command Prompt

To stop Oracle services from the command prompt, enter:

```
C:\> net STOP service
```

where *`service`* is a specific service name, such as `OracleServiceORCL`.

Oracle Administration Assistant for Windows NT

To stop Oracle services from Oracle Administration Assistant for Windows NT:

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Administration Assistant for Windows NT.
2. Right-click the *`sid`*.

   where *`sid`* is a specific instance name, such as `orcl`.
3. Click Stop Service

   This stops service `OracleServiceORCL`.

Auto-starting Oracle Services

Oracle services can be set to start automatically whenever the Windows computer is restarted. You can turn auto-start on or off from two different locations:

- Control Panel
- Oracle Administration Assistant for Windows NT

Control Panel

To use the Control Panel to configure when and how Oracle9*i* database is started:

1. Choose Start > Settings > Control Panel.

   The Control Panel window appears.
2. Double-click Services.

   The Services dialog box appears.

> **Note:** If you are using Windows 2000, then you will not see Services in the Control Panel. Double-click Administrative Tools first, then double-click Services.

3. Select service `OracleServiceSID` and click Startup.

   The Service dialog box appears.

4. Choose Automatic from the Startup Type field.
5. Click OK.
6. Click Close to exit the Services dialog box.

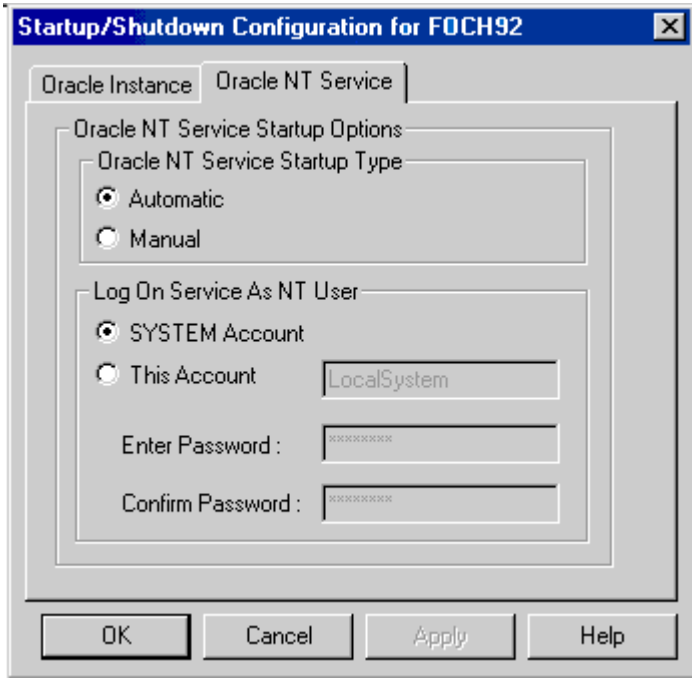
Oracle Administration Assistant for Windows NT

To automatically start Oracle services from Oracle Administration Assistant for Windows NT:

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Administration Assistant for Windows NT.
2. Right-click the *`sid`*.

   where *`sid`* is a specific instance name, such as `orcl`.

3. Choose Startup / Shutdown Options.
4. Choose the Oracle NT Service tab.
5. Choose Automatic in the Oracle NT Service Startup Type box.
6. Click Apply.
7. Click OK.



Starting and Shutting Down a Database with SQL*Plus

These instructions assume that a database instance has been created.

> **Note:** Directory path examples in this chapter follow **Optimal Flexible Architecture (OFA)** guidelines (for example, *ORACLE_BASE**ORACLE_HOME*\rdbms\admin). If you specified non-OFA compliant directories during installation, then your directory paths will differ. See *Oracle9i Database Getting Started for Windows* for more information.

To start or shut down an Oracle9*i* database:

1. Go to your Oracle9*i* database server.
2. Start SQL*Plus at the command prompt:

```
C:\> sqlplus / NOLOG
```

3. Connect to Oracle9*i* database with **username SYSDBA**:

```
SQL> CONNECT / AS SYSDBA
```

4. To start a database, enter:

```
SQL> STARTUP [PFILE=path\filename]
```

This command uses the **initialization parameter file** specified in *path**filename*. To start a database using a file named

```
init2.ora in C:\ora92\admin\orcl\pfile
```

for example, you would enter:

```
SQL> STARTUP PFILE=C:\ora92\admin\orcl\pfile\init2.ora
```

If no PFILE is specified, then the command uses the default initialization parameter file located in directory *ORACLE_BASE*\ADMIN\db_name\pfile.

5. To stop a database, enter:

```
SQL> SHUTDOWN [mode]
```

where mode is normal, immediate, or abort.

In a normal shutdown, Oracle9*i* database waits for all currently-connected users to disconnect and disallows any new connections before shutting down. This is the default mode.

In an immediate shutdown, Oracle9*i* database terminates and rolls back active transactions, disconnects clients, and shuts down.

In an abort shutdown, Oracle9*i* database terminates active transactions and disconnects users; it does not roll back transactions. The database performs automatic **recovery** and rollback the next time it is started. Use this mode only in emergencies.

> **See Also:** *Oracle9i Database Getting Started for Windows* for a list of other tools that can start Oracle9*i* database and this guide for information on options you can specify when starting your database.

Starting and Shutting Down a Database Using Services

You can start or shut down Oracle9*i* database by starting or stopping service `OracleServiceSID` in the Control Panel. Starting `OracleServiceSID` is equivalent to using the `STARTUP` command or manually entering:

```
C:\> oradim -STARTUP -SID SID -USERPWD password -STARTTYPE srv,inst
```

Stopping `OracleServiceSID` is equivalent to using the `SHUTDOWN` command or manually entering:

```
C:\> oradim -SHUTDOWN -SID SID -USERPWD password -SHUTTYPE srv,inst -SHUTMODE -i
```

You can enable starting and stopping Oracle9*i* database through `OracleService`*`SID`* two different ways:

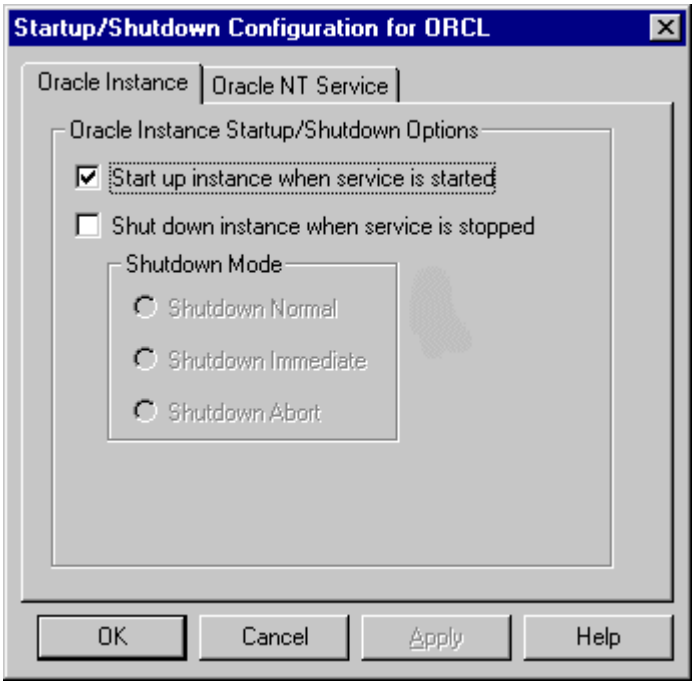
- Oracle Administration Assistant for Windows NT
- Setting Registry Parameters

Oracle Administration Assistant for Windows NT

To start or stop a database using Oracle Services from Oracle Administration Assistant for Windows NT:

1. Choose Start > Programs > Oracle - *`HOME_NAME`* > Configuration and Migration Tools > Administration Assistant for Windows NT.
2. Right-click the *`SID`*.

   where *`SID`* is a specific instance name, such as `ORCL`.
3. Choose Startup / Shutdown Options.
4. Choose the Oracle Instance tab.
5. Select Start up instance when service is started, Shut down instance when service is stopped, or both.



Setting Registry Parameters

To start or stop an Oracle9*i* database through Oracle Services, set the following registry parameters to the indicated values:

- ORA_*SID*_AUTOSTART

  When set to `true`, the default value, this parameter causes Oracle9*i* database to start when `OracleService`*`SID`* is started.

- ORA_*SID*_PFILE

  This parameter sets the full path to the initialization parameter file. The default path is *`ORACLE_BASE`*`\ADMIN\db_name\pfile\init.ora`

- ORA_SHUTDOWN

  When set to `true`, this parameter enables the selected Oracle9*i* database to be shut down when `OracleService`*`SID`* is stopped. This includes any database in the current Oracle home. Default value is `false`.

- ORA_*SID*_SHUTDOWN

When set to `true`, this parameter causes the Oracle9*i* database identified by the *`SID`* value to shut down when `OracleService`*`SID`* is stopped. Default value is `false`.

Caution: If `ORA_SHUTDOWN` or `ORA_SID_SHUTDOWN` is set to `false`, then shutting down `OracleServiceSID` will still shut down Oracle9*i* database. But it will be an abnormal shutdown, and Oracle Corporation does not recommend it.

The following two registry parameters are optional:

- `ORA_`*`SID`*`_SHUTDOWNTYPE`

  This parameter controls database shutdown mode. Set it to `a` (`abort`), `i` (`immediate`), or `n` (`normal`). Default mode is `i` (`immediate`) if you do not set this parameter.

- `ORA_`*`SID`*`_SHUTDOWN_TIMEOUT`

  This parameter sets maximum time to wait before the service for a particular *`SID`* stops.

The registry location of these required and optional parameters is determined by the number of Oracle home directories on your computer. If you have only one Oracle home directory, then these parameters belong in:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME0
```

If you have multiple Oracle home directories, then these parameters belong in:

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

where *ID* is incremented for each additional Oracle home directory on your computer.

Note: If you use ORADIM utility to create or edit instances, then the utility automatically sets the relevant registry parameters to their appropriate values.

See Also: For instructions on adding and editing registry parameters, see *Oracle9i Database Getting Started for Windows*

Starting or Stopping OracleServiceSID from the Control Panel

1. Choose Start > Settings > Control Panel.

   The Control Panel window appears.

2. Double-click Services.

   The Services dialog box appears.

3. To start the database, select `OracleService`*`SID`* and choose Start.

   This automatically starts ORADIM utility and issues the `-STARTUP` command using the **initialization parameter file** identified by `ORA_`*`SID`*`_PFILE`.

4. To stop the database, select `OracleService`*`SID`* and choose Stop.

   This automatically starts ORADIM utility, which issues the `-SHUTDOWN` command in the mode indicated by `ORA_`*`SID`*`_SHUTDOWNTYPE`, and shuts down your Oracle9*i* database.

Running Multiple Instances

To run multiple instances:

1. Start the service for each instance using ORADIM utility or the Services dialog box of the Control Panel.
2. At the command prompt set the ORACLE_SID configuration parameter to the *SID* for the first instance to run:

   ```
   C:\> SET ORACLE_SID=SID
   ```

   where *SID* is the name of the Oracle9*i* database instance.
3. Start SQL*Plus:

   ```
   C:\> sqlplus / NOLOG
   ```

4. Connect AS SYSDBA:

   ```
   SQL> CONNECT / AS SYSDBA
   ```

5. Start up the first instance:

   ```
   SQL> STARTUP PFILE=ORACLE_BASE\admin\db_name\pfile\init.ora
   ```

   where *ORACLE_BASE* is c:\oracle (unless you changed it during installation) and *db_name* is the name of the instance.
6. Repeat Steps 2-5 for the other instances to run.

Creating and Populating Password Files

Use Password Utility to create password files. Password Utility is automatically installed with Oracle9*i* Utilities. Password files are located in directory *ORACLE_BASE**ORACLE_HOME*\DATABASE and are named PWD*sid*.ORA, where *SID* identifies the Oracle9*i* database instance. Password files can be used for local or remote connections to an Oracle9*i* database.

To create and populate a password file:

1. Create a password file with Password utility:

   ```
   C:\> orapwd FILE=PWDsid.ora PASSWORD=password ENTRIES=max_users
   ```

   where

   - FILE specifies the password filename.
   - *SID* identifies the database instance.
   - PASSWORD sets the password for account SYS.
   - ENTRIES sets maximum number of entries in password file. This corresponds to maximum number of distinct users allowed to connect to the database simultaneously with either the **SYSDBA** or the **SYSOPER** DBA **privilege**.

2. Set initialization parameter file parameter REMOTE_LOGIN_PASSWORDFILE to exclusive, shared, or none.

   exclusive specifies that only one instance can use the password file and that the password file contains names other than SYS. In search of the password file, Oracle9*i* looks in the registry for the value of parameter ORA_SID_PWFILE. If no value is specified, then it looks in the registry for the value of parameter ORA_PWFILE, which points to a file containing usernames, passwords, and privileges. If that is not set, then it uses the default:

   ```
   ORACLE_BASE\ORACLE_HOME\DATABASE\PWDsid.ORA.
   ```

   shared is the default value. It specifies that multiple instances (for example, a Real Application Clusters environment) can use the password file. However, the only user recognized by the password file is SYS. Other users cannot log in with SYSOPER or SYSDBA privileges even if those privileges are granted in the password file. The shared value of this parameter affords backward compatibility with earlier Oracle database software releases. Oracle9*i* database looks for the same files as it does when the value is exclusive.

   none specifies that Oracle9*i* database ignores the password file and that **authentication** of privileged users is handled by the Windows operating system. This is the default setting.

3. Start SQL*Plus:

   ```
   C:\> sqlplus / NOLOG
   ```

4. Connect AS SYSDBA:

```
SQL> CONNECT / AS SYSDBA
```

5. Start Oracle9*i* database:

```
SQL> STARTUP
```

6. Grant appropriate privileges to each user. Users who must perform database administration, for example, would be granted privilege SYSDBA:

```
SQL> GRANT SYSDBA TO scott;
```

   If the grant is successful, then the following message displays:

```
Statement Processed.
```

   This adds `scott` to the password file and enables `scott` to connect to the database with SYSDBA privileges. Use SQL*Plus to add or delete usernames, user passwords, and user privileges in password files.

> **Caution:** Copying or manually moving password files may result in ORADIM utility being unable to find a password to start an instance.

Viewing and Hiding the Password File

The password file is not automatically hidden. It can be made invisible and visible again from two different locations:

- Command Prompt
- Windows NT Explorer

> **Note:** The password file must be visible before it can be moved, copied, or deleted.

Command Prompt

1. To see the password file, enter:

   ```
   C:\oracle\ora92\database> attrib
   ```

   The password file is displayed as PWD*sid*.ora:

   ```
   A       C:\oracle\ora92\database\oradba.exe
   A       C:\oracle\ora92\database\OraDim.Log
   A       C:\oracle\ora92\database\PWDsid.ora
   A       C:\oracle\ora92\database\SPFILEsid.ora
   ```

2. To make the password file invisible, enter:

   ```
   C:\oracle\ora92\database> attrib +H PWDsid.ora
   ```

3. To see the effect of the change, enter:

   ```
   C:\oracle\ora92\database> attrib
   ```

   The password file is now hidden:

   ```
   A       C:\oracle\ora92\database\oradba.exe
   A       C:\oracle\ora92\database\OraDim.Log
   A   H   C:\oracle\ora92\database\PWDsid.ora
   A       C:\oracle\ora92\database\SPFILEsid.ora
   ```

4. To make the password file visible again, enter:

   ```
   C:\oracle\ora92\database> attrib -H PWDsid.ora
   ```

Windows NT Explorer

To make the password file invisible or visible again:

1. Navigate to directory C:\oracle\ora92\database.
2. Right-click PWD*sid*.ora.
3. Choose Properties.

   The PWD*sid*.ora Properties dialog box opens

4. In Attributes, check or clear the checkbox next to Hidden.
5. Click OK.

To view or hide an invisible password file:

1. Navigate to directory `C:\oracle\ora92\database`.
2. Choose Folder Options from the View main menu.
3. Choose the View tab.
4. To view an invisible password file, choose Show hidden files and folders.
5. To hide an invisible password file, choose Do not show hidden files and folders.
6. Click OK.

Connecting Remotely to the Database as SYS

When connecting to the **starter database** from a **remote computer** as SYS, you must use a different password from the one described in *Oracle9i Database Installation Guide for Windows* when logging on with SYSDBA privileges. This is because the password file enables database access in this situation and it requires the password `oracle` for this purpose.

Encrypting Database Passwords

With Oracle9*i* database, you can encrypt the password used to verify a **remote database** connection.

To enable password encryption:

1. Add `DBLINK_ENCRYPT_LOGIN` to the initialization parameter file on the server computer.
2. Set `DBLINK_ENCRYPT_LOGIN` equal to `true`.
3. Set configuration variable `ORA_ENCRYPT_LOGIN` on the client computer to `true`. See *Oracle9i Database Getting Started for Windows* for instructions on adding and setting configuration parameters in the registry.

Once these parameters are set to `true`, whenever a user attempts a remote login, Oracle9*i* database encrypts the password before sending it to the remote database. If the connection fails, then the failure is noted in the audit log. Oracle9*i* database then checks if either of these parameters is set to `false`. If so, Oracle9*i* database attempts the connection again using an unencrypted version of the password. If the connection is successful, then the success is noted in the audit log, and the connection proceeds.

Note: Oracle database software releases before 7.1 do not support encrypted passwords. If you are connecting to an earlier version of the Oracle database, then you must set initialization parameter DBLINK_ENCRYPT_LOGIN to false for the connection to succeed.

Creating Control, Data, and Log Files on Remote Computers

Oracle can access database files on a **remote computer** using Universal Naming Convention (UNC), but it may degrade database performance and network reliability. UNC is a PC format for specifying locations of resources on a local area network. UNC uses the following format:

```
\\server-name\shared-resource-path-name
```

For example, UNC specification for file system01.dbf in directory C:\oracle\oradata\orcl on shared server argon would be:

```
\\argon\oracle\oradata\orcl\system01.dbf
```

Locations of archive log files cannot be specified using UNC. Always set initialization parameter LOG_ARCHIVE_DEST_*n* to a mapped drive. If you set it to a UNC specification, then Oracle9*i* database does not start and you receive the following errors:

```
ORA-00256: error occurred in translating archive text string '\meldell\rmdrive'
ORA-09291: sksachk: invalid device specified for archive destination
OSD-04018: Unable to access the specified directory or device
O/S-Error: (OS 2) The system cannot find the file specified
```

An ORA-00256 error also occurs if you enter \\\meldell\rmdrive or \\\meldell\\rmdrive. Control files required the additional backslashes for Oracle8 release 8.0.4, but **redo log files** and datafiles did not.

Archiving Redo Log Files

If you installed your Oracle9*i* database through the Typical installation, then it is created in NOARCHIVELOG mode. If you created your database through the Custom option of Database Configuration Assistant, then you had the choice of either ARCHIVELOG or NOARCHIVELOG.

In NOARCHIVELOG mode, redo logs are not archived. Setting your archive mode to ARCHIVELOG and enabling automatic archiving causes **redo log files** to be archived. This protects Oracle9*i* database from both instance and disk failure.

This section describes how to change archive mode to ARCHIVELOG and enable automatic archiving.

> **See Also:** For complete descriptions of ARCHIVELOG and NOARCHIVELOG modes, see "Managing Archived Redo Logs" of *Oracle9i Database Administrator's Guide*

Changing Archive Mode to ARCHIVELOG

To change archive mode to ARCHIVELOG:

1. Start SQL*Plus at the command prompt:

   ```
   C:\> sqlplus / NOLOG
   ```

2. Connect to the Oracle9*i* database as SYSDBA:

   ```
   SQL> CONNECT / AS SYSDBA
   ```

3. If the database is open, then shut it down:

   ```
   SQL> SHUTDOWN
   ```

4. Next, **mount** the database:

   ```
   SQL> STARTUP MOUNT
   ```

5. Enter the following command:

   ```
   SQL> ARCHIVE LOG LIST
   ```

   The following output indicates the database is not in archive mode:

   ```
   Database log mode              No Archive Mode
   Automatic archival             Disabled
   Archive destination            %RDBMS%\
   Oldest online log sequence     34
   Current log sequence           37
   ```

6. Change archive mode to ARCHIVELOG:

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

7. Enter the following command:

```
SQL> ARCHIVE LOG LIST
```

The following output indicates the database is now in archive mode:

```
Database log mode              Archive Mode
Automatic archival             Disabled
Archive destination            %RDBMS%\
Oldest online log sequence     34
Current log sequence           37
```

8. Open the database:

```
SQL> ALTER DATABASE OPEN;
```

Enabling Automatic Archiving

To enable automatic archiving:

1. Open file *ORACLE_BASE*\ADMIN*db_name*\pfile\init.ora.
2. Find the following three initialization parameters:

```
# LOG_ARCHIVE_START = true
# LOG_ARCHIVE_DEST_1 = %ORACLE_HOME%\database\archive
# LOG_ARCHIVE_FORMAT = "%%ORACLE_SID%%T%TS%S.ARC"
```

3. Remove the # sign from in front of each.

> **Note:** Double quotation marks around LOG_ARCHIVE_FORMAT

4. Edit the LOG_ARCHIVE_DEST_*n* value to identify an existing drive and directory in which to archive your filled redo logs.

> **Note:** Default size of Redo logs is approximately 100 MG.

5. Edit the LOG_ARCHIVE_FORMAT value to indicate the appropriate archiving format:

   %%*ORACLE_SID*%%%T.ARC specifies the **thread** number. This number is padded to the left by zeroes. The default value is one with a range of up to three characters. Example: SID0001.ARC

   %%*ORACLE_SID*%%%S.ARC specifies the log sequence number. This number is padded to the left by zeroes. The default value is one with a range of up to five characters. Example: SID0001.ARC

   %%*ORACLE_SID*%%%t.ARC specifies the thread number. The number is not padded. The default value is one with no range limit on characters. Example: SID1.ARC

   %%*ORACLE_SID*%%%s.ARC specifies the log sequence number. The number is not padded. The default value is one with no range limit on characters. Example: SID1.ARC

6. Save your changes.

7. Exit the file.

8. Shut down the database:

   ```
   SQL> SHUTDOWN
   ```

9. Restart the database

   ```
   SQL> STARTUP
   ```

10. Enter the following command:

    ```
    SQL> ARCHIVE LOG LIST
    ```

    The following output indicates that automatic archiving of redo log files is enabled and an archiving destination is specified:

    ```
    Database log mode              Archive Mode
    Automatic archival             Enabled
    Archive destination            C:\BACKUP
    Oldest online log sequence     34
    Current log sequence           37
    ```

Using ORADEBUG Utility

ORADEBUG utility is a debugging tool that sends debug commands through SQL*Plus to an Oracle **process**. It is primarily for use by developers and Oracle Support Services personnel. Use this utility only when instructed to do so by Oracle Support Services. You must have database administrator privileges to use ORADEBUG utility.

To start ORADEBUG utility:

1. Start SQL*Plus from the command prompt:

   ```
   C:\> sqlplus / NOLOG
   ```

2. Connect to Oracle9*i* database as SYSDBA:

   ```
   SQL> CONNECT / AS SYSDBA
   ```

3. Enter the following at the SQL*Plus prompt:

   ```
   SQL> ORADEBUG
   ```

   ORADEBUG utility runs and prompts you for parameters. To obtain a list of these parameters, enter the following at the SQL*Plus prompt:

   ```
   SQL> ORADEBUG HELP
   ```

   Output from most debug commands is written to a **trace file**. Trace files are created in the directory specified by initialization parameters BACKGROUND_DUMP_DEST and USER_DUMP_DEST. By default, these parameters are set to *ORACLE_BASE*\ADMIN*db_name*\bdump and *ORACLE_BASE*\ADMIN*db_name*\udump, respectively. To find the location of your trace file, enter the following at the SQL*Plus prompt:

   ```
   SQL> ORADEBUG TRACEFILE_NAME
   ```

   If output from a debug command produces more than one line of output, then the result is sent to a trace file, and a message indicating that the command has completed is relayed to SQL*Plus. If output from a debug command produces only one line of output, then the output is relayed directly to SQL*Plus.

   Note: There is currently a limitation when using ORADEBUG utility. If you attempt to debug a thread that is blocking on I/O, ORADEBUG can cause SQL*Plus to hang until that I/O completes.

4

Monitoring a Database

This chapter describes how to monitor Oracle9*i* for Windows.

This chapter contains these topics:

- Database Monitoring Overview
- Using Oracle for Windows NT Performance Monitor
- Using Event Viewer
- Trace Files
- Alert Files
- Viewing Threads

Database Monitoring Overview

Table 4–1 describes tools that enable you to monitor your Oracle9*i* database:

Table 4–1 Database Monitoring Tools

Tool	Functionality
Oracle for Windows NT Performance Monitor	Monitor database objects, such as CPU usage, buffer cache, or a background **process**.
Event Viewer	Monitor database events.
Trace Files	Record occurrences and exceptions of database operations
Alert Files	Record important information about error messages and exceptions during database operations.
Oracle Enterprise Manager Database Management Packs	Monitor and tune using tools with real-time graphical performance information. **See Also**: Your Oracle Enterprise Manager documentation set for more information
Oracle Administration Assistant for Windows NT	View information on or terminate any Oracle **thread**.

See Also: *Oracle9i Database Performance Guide and Reference* for general tuning information

Using Oracle for Windows NT Performance Monitor

Oracle for Windows NT Performance Monitor is a graphical tool for measuring the performance of Oracle9*i* for Windows objects on a local server or other servers on a network. This tool is the same in appearance and operation as Windows NT Performance Monitor, except it has been preloaded with Oracle9*i* database performance elements.

On each computer, you can view behavior of objects, such as buffer cache, **data dictionary** cache, datafiles, threads, and processes. An object is a graphical representation of an element in your system. Every element, resource, and device in your system can be represented as an object.

A set of counters is associated with each object. A counter is a unit of measurement used by Performance Monitor to display activity. The type of activity the counter measures is dependent upon type of object.

Certain types of objects and their respective counters are present on all systems. Other counters, such as application-specific counters, appear only if the computer is running the associated software.

Each of these objects has an associated set of counters that provide information about device usage, queue lengths, delays, and information used to measure throughput and internal congestion.

Registry Information

When you install Oracle for Windows NT Performance Monitor, values are automatically set in the **registry** as described in "Configuration Parameters and the Registry" of *Oracle9i Database Getting Started for Windows* guide.

Oracle for Windows NT Performance Monitor lets you monitor only one database **instance** at a time. For this reason, the registry contains the following values:

- `Hostname`
- `Username`
- `Password`

Use OPERFCFG utility to change these values. Oracle Corporation recommends setting the security level on each of these registry values.

> **See Also:** "Configuration Parameters and the Registry" in *Oracle9i Database Getting Started for Windows* for instructions on how to use OPERFCFG utility.

To use Oracle for Windows NT Performance Monitor for another database instance on the same computer or a UNIX computer, change the values appropriately in the registry. You can also monitor non-Windows NT Oracle databases by changing `Hostname` registry value so it points to another computer specified in file `tnsnames.ora`.

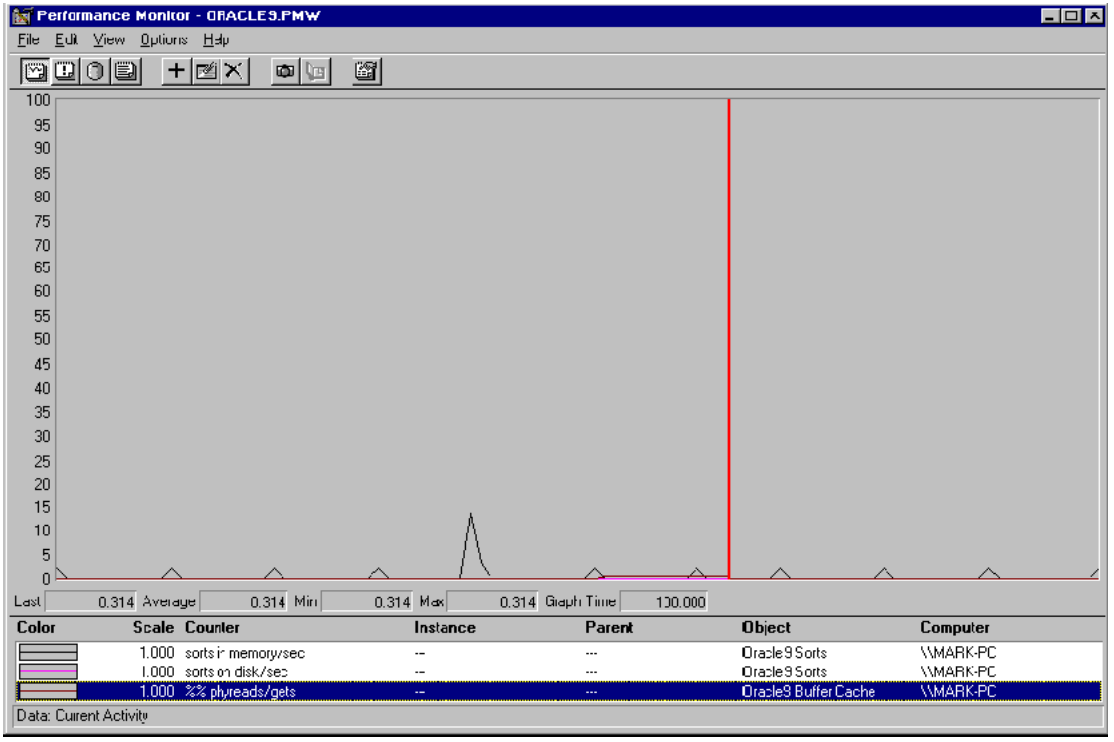
Accessing Oracle for Windows NT Performance Monitor

To access Oracle for Windows NT Performance Monitor:

Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Oracle for Windows NT Performance Monitor.

Figure 4–1 shows Performance Monitor displaying Chart View:

Figure 4–1 Oracle Performance Monitor Window



Oracle for Windows NT Performance Monitor has four views you can choose from the View menu:

- Chart view displays database activity as it occurs.
- Alert view lets you know when certain minimum performance criteria are not being met, or maximum criteria are being exceeded.

- Log view maintains continuous records on performance.
- Report view saves information about specific criteria.

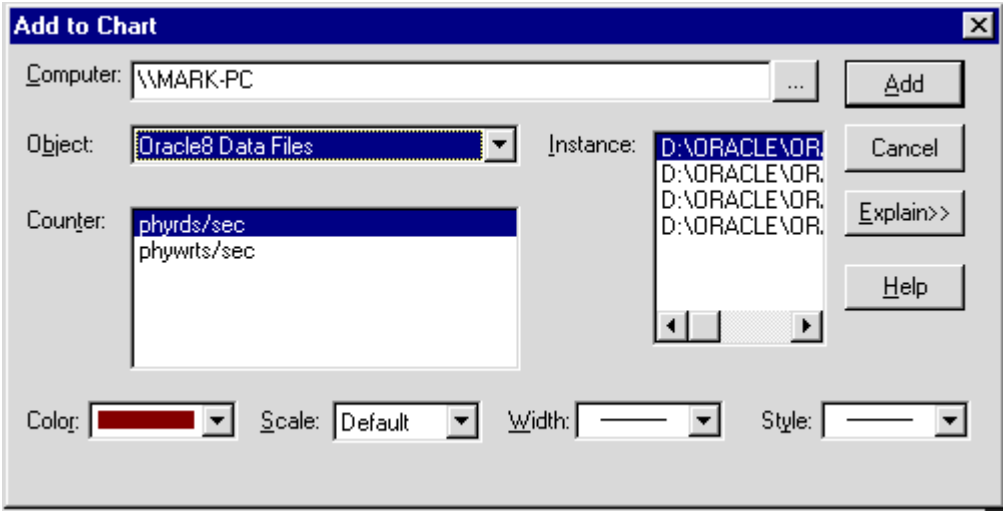
> **See Also:** Your Windows NT documentation for complete information about the four views

Modifying Views

For each view you can decide objects to monitor, what counters to use for each object, and graphical attributes of the view. When an object is chosen, it is assigned a counter, a color, and added to the status bar at the bottom of Oracle for Windows NT Performance Monitor.

To add objects to a view:

1. Choose Add To (Chart, Alert, Log, Report) from the Edit menu. (This example uses Add to Chart, but other dialogs are similar.)



   The Add to Chart dialog box appears.

2. In the Computer list box, select the computer to monitor.
3. In the Object list box, select the object or objects to monitor, then choose Add.

   If no data or Oracle9*i* objects appear, then either Oracle9*i* database is not running or an invalid host string or password has been entered. If the database

is not started, then exit Oracle for Windows NT Performance Monitor, start the database, and restart Oracle for Windows NT Performance Monitor.

4. In the Counter list box, select one or more counters for each object you have selected. Note that the contents of the Counter box change depending upon your selection in the Object box. For details on how a counter works, highlight the counter and choose Explain.
5. In the Instance box, select an instance for this counter.
6. In the Color box, choose a color for the display of the selected counter.
7. In the Scale box, choose the scale at which to display the counter.
8. In the Width box, specify the width of the line on the graph.
9. In the Style box, choose a style for your graph line.
10. Click Done when you are finished.

    The selections you have chosen to monitor are displayed.

Understanding Oracle Performance Objects

All Oracle9*i* system resources that can be monitored through Oracle for Windows NT Performance Monitor begin with Oracle9*i*. This section lists Oracle9*i* objects and describes their associated counters. These measures are defined in

*ORACLE_BASE**ORACLE_HOME*\dbs\perf.ora

For additional information on these objects, see *Oracle9i Database Performance Guide and Reference*.

Note: You can monitor only one instance at a time using Oracle for Windows NT Performance Monitor on a given computer.

Oracle9*i* Buffer Cache

The counter is phyrds / gets %. The percentage of phyrds / gets is calculated as a Miss ratio. The lower the Miss counter, the better. To improve performance, increase the number of buffers in the buffer cache, if memory is available on the computer. This value is not time-derived.

Oracle9*i* Redo Log Buffer

The counter is redo log space requests. The value of this counter must be near zero. If this value increments consistently, then processes have had to wait for space in the **redo log buffer**. In this case, it may be necessary to increase the size of the redo log buffer.

Oracle9*i* Data Dictionary Cache

The counter is getmisses / gets %. The value of this counter must be less than 10 or 15% for frequently accessed data dictionary caches. If the ratio continues to increase over this threshold while your application is running, then increase the amount of memory available to the data dictionary cache.

To increase the memory available to the cache, increase the value of initialization parameter SHARED_POOL_SIZE. (See *Oracle9i Database Performance Guide and Reference* for more detailed information on tuning memory allocation in Oracle9*i* database.) This value is not time-derived.

Oracle9*i* Library Cache

The counter is reloads / pins %. This is the percentage of SQL statements, **PL/SQL** blocks, and object definitions that required reparsing. Total Reloads must be near zero. If the ratio of Reloads to Pins is greater than 1%, then reduce the library cache misses. This value is not time-derived.

Oracle9*i* Data Files

The counters are phyrds / sec and phywrts / sec. These values are time-derived. Disk contention occurs when multiple processes try to access the same disk simultaneously. Depending on results from monitoring disk activity, corrective actions could include:

- Distributing I / O
- Separating datafiles and **redo log files**
- Separating tables and indexes
- Striping table data

Oracle9*i* DBWR stats1

The two counters available, buffers scanned / sec and LRU scans / sec, are helpful in tuning Buffer Cache. Buffers scanned / sec is the number of buffers DBWR scanned in each second. The buffers scanned are on the LRU (Least Recently Used) list. LRU

scans / sec is the number of times DBWR scanned the (Least Recently Used) buffer list in each second.

Oracle9*i* DBWR stats2

The two counters available, timeouts / sec and checkpoints / sec, are helpful in determining how much work DBWR has been requested to perform. Timeouts / sec is the number of times DBWR timed-out in each second. DBWR is on a three second timeout interval. If DBWR has not been posted within a three second interval, then it times out.

Checkpoints / sec is the number of checkpoint messages processed by database writer in each second. Whenever a checkpoint occurs, DBWR must be messaged (posted) to "write dirty buffers to disk".

Oracle9*i* Dynamic Space Management

The counter is recursive calls / sec. Dynamic extension causes Oracle9*i* to execute SQL statements in addition to those SQL statements issued by user processes. These SQL statements are called recursive calls.

If Oracle9*i* makes excessive recursive calls while an application is running, then it may be necessary to determine the cause. Examine the recursive calls statistic through dynamic performance table V$SYSSTAT.

Oracle9*i* Free List

The counter is free list waits / requests %. Contention for free lists is reflected by contention for free data blocks in buffer cache. You can determine if contention for free lists is reducing performance by querying V$WAITSTAT.

If the number of free list waits for free blocks is greater than 1% of the total number of requests, then consider adding more free lists to reduce contention.

Oracle9*i* Sorts

The available counters are sorts in memory / sec and sorts on disk / sec. The default sort area size is adequate to hold all data for most sorts. However, if your application often performs large sorts on data that does not fit into the sort area, then you may increase sort area size.

Oracle for Windows NT Performance Monitor Troubleshooting Information

If no data or Oracle9*i* objects appear in the Objects list of the Add to (Chart, Alert, Log, Report) dialog box, then:

1. Ensure that Oracle9*i* database is running. If it is not running, then exit Oracle for Windows NT Performance Monitor, and start the database.
2. If the database is running but no data or Oracle9*i* objects appear, then check the error file for Oracle for Windows NT Performance Monitor:

   ```
   ORACLE_BASE\ORACLE_HOME\dbs\operf.log
   ```

   If the log file indicates an invalid host string or password, then check the registry for correct values for `Hostname`, `Password`, and `Username`. See "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet" in *Oracle9i Database Getting Started for Windows* for further information about these values.
3. Restart Oracle for Windows NT Performance Monitor.

Using Event Viewer

Oracle9*i* for Windows problems and other significant occurrences are recorded as events in an application event log. View and manage these recorded events in Event Viewer.

To access Event Viewer:

1. Choose Start > Programs > Administrative Tools > Event Viewer.

   The Event Viewer window appears.
2. Choose Application from the Log menu.

Figure 4–2 displays the Application view window, Table 4–2 shows what is recorded in each column, and Table 4–3 interprets icons that appear at the left edge of the viewer.

Figure 4–2 Application View Window

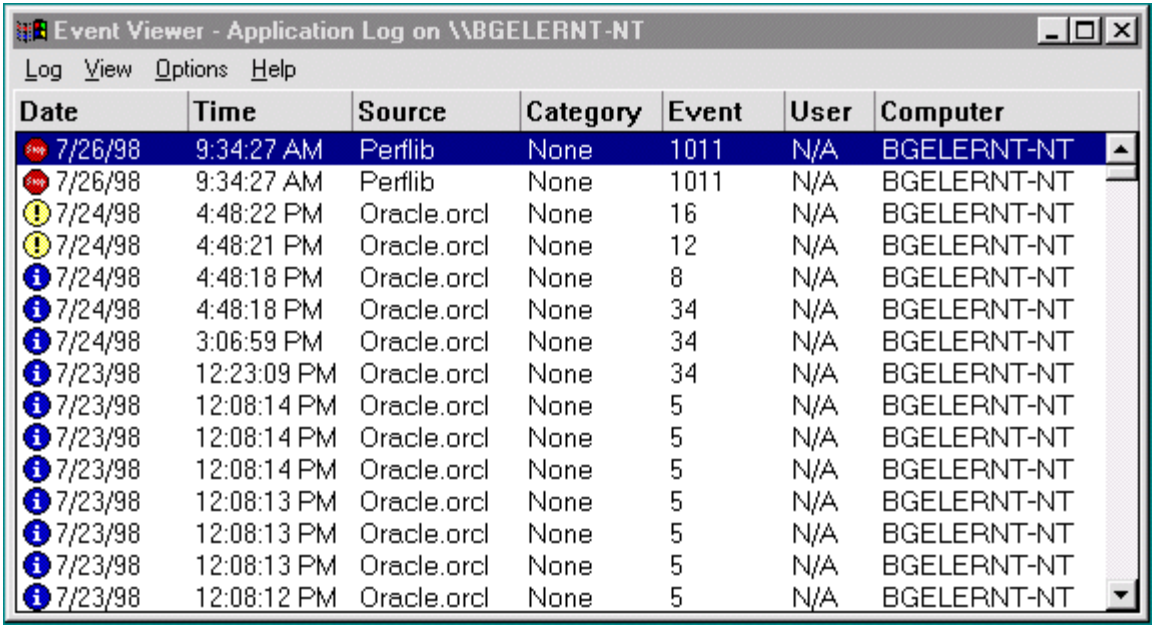


Table 4–2 Application View Definitions

Column Name	Definition
Date	Date event took place
Time	Time event took place
Source	Application that recorded event
Category	Classification of event
Event	Unique number assigned to event
Computer	Computer name on which event occurred

Table 4–3 Event Viewer Icons

Icon	Event Type	Suggested Action
Red Stop Sign	Error	Always check these icons.
Lower-case "i" in Blue Circle	Information	Noncritical system events. Check these icons only to track a specific event.
Exclamation Point in Yellow Circle	Warning	Special events, such as instance termination or services shutdown. Investigate these icons, but they are usually noncritical.

Reading Event Viewer

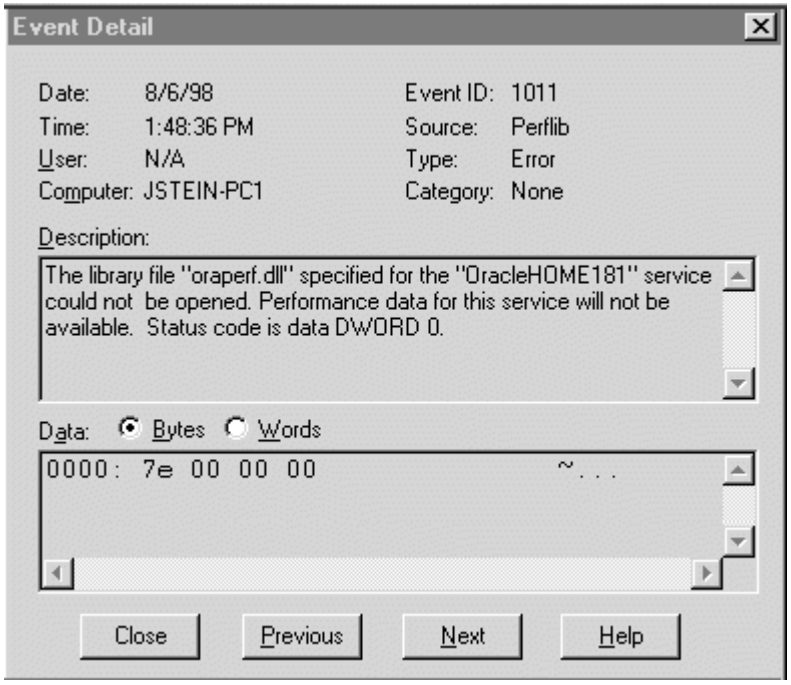
Oracle9*i* for Windows NT events display with a source of Oracle.orcl.

Event number 34 specifies an audit trail event. These events are recorded if parameter `AUDIT_TRAIL` is set to `db` (`true`) or `os` in the initialization parameter file. Option `os` enables system wide auditing and causes audited records to be written to Event Viewer. Option `db` enables system wide auditing and causes audited records to be written to the database audit trail (table `SYS.AUD$`). Some records, however, are written to Event Viewer.

Event numbers other than 34 specify general database activities, such as an instance being started or stopped.

When you double-click an icon in Event Viewer, the Event Detail dialog appears with more information about the selected event. Figure 4–3, for example, shows details about EventID 1011. In the Description text box you will find a text description of the event. In the Data text box you can choose Bytes to see the information in hexadecimal format or Words to see DWORDS for the same data.

Figure 4–3 Event Detail Dialog



See Also: Microsoft Windows documentation for more information on using Event Viewer

Managing Event Viewer

Setting `AUDIT_TRAIL` to `db` or `os` causes more records to be written to Event Viewer. This can fill up the Event Viewer log file. Follow these procedures to increase log file size:

1. Choose Log Settings from the Log menu.

   The Event Log Settings dialog box appears.

2. Adjust the setting in the Maximum Log Size field to an appropriate level.

3. Click OK.

   You are returned to Event Viewer.

Caution: Audit information cannot be spooled to a file. Parameter AUDIT_FILE_DEST is not supported in Windows and should not be added to the initialization parameter file.

Trace Files

Oracle9*i* for Windows background threads use trace files to record occurrences and exceptions of database operations, as well as errors. Background thread trace files are created regardless of whether parameter BACKGROUND_DUMP_DEST is set in the **initialization parameter file**. If BACKGROUND_DUMP_DEST is set, then trace files are stored in the directory specified. If the parameter is not set, then trace files are stored in directory *ORACLE_BASE*\ADMIN*db_name*\bdump.

Oracle9*i* database creates a different **trace file** for each background thread. The name of the trace file contains the name of the background thread, followed by the extension .TRC. Sample trace file syntax includes:

- SIDDBWR.TRC
- SIDSMON.TRC

where *SID* is the name of the instance.

Trace files are also created for user threads if parameter USER_DUMP_DEST is set in the initialization parameter file. Trace files for user threads have the form ora*xxxxx*.trc, where *xxxxx* is a 5-digit number indicating the Windows thread ID.

Alert Files

Alert files contain important information about error messages and exceptions that occur during database operations. Each Oracle9*i* for Windows 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.

For example, when automatic archiving of redo logs is halted because no disk space is available, a message is placed in the alert file. The alert file is the first place to check if something goes wrong with the database and the cause is not immediately obvious.

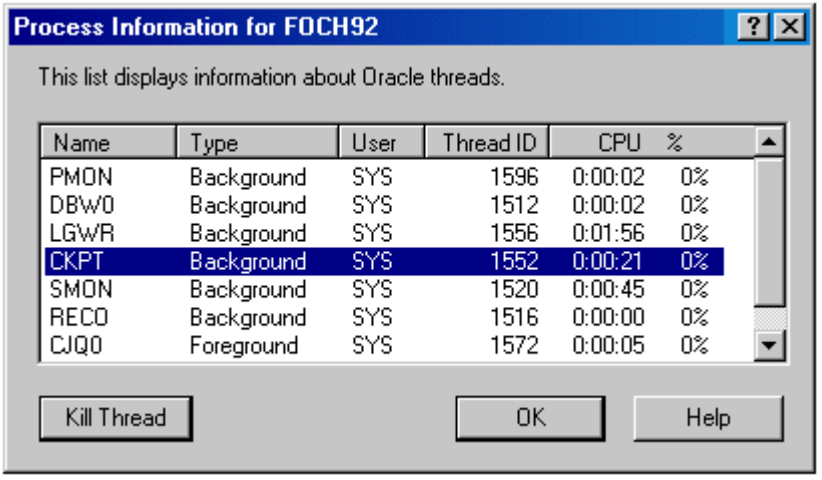
The alert file is named *SID*ALRT.LOG and is found in the directory specified by parameter BACKGROUND_DUMP_DEST in the initialization parameter file. If parameter BACKGROUND_DUMP_DEST is not set, then file *SID*ALRT.LOG is generated in *ORACLE_BASE*\admin*db_name*\bdump. Alert files should be deleted or archived periodically.

Viewing Threads

To view information on Oracle threads using Oracle Administration Assistant for Windows NT:

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Administration Assistant for Windows NT.
2. Right-click the *SID*, where *SID* is a specific instance name, such as orcl.
3. Choose Process Information.

   The Process Information dialog appears, listing name, type, user, thread ID, and CPU usage for each Oracle thread.
4. To terminate a thread, select it and choose Kill Thread.



5

Tuning Windows NT to Optimize Oracle9*i*

This chapter describes how to tune the Windows NT Server operating system to ensure that your Oracle9*i* database is running in the best possible environment.

This chapter contains these topics:

- Windows NT Tuning Overview
- Reduce Priority of Foreground Applications on Server Console
- Configure Windows NT Server to Be an Application Server
- Disable Unnecessary Services
- Remove Unused Network Protocols
- Reset Network Protocol Bind Order
- Apply Latest Reliable Windows NT Server Service Pack
- Use Hardware or Operating System Striping
- Multiplex Windows NT Server Virtual Memory Paging File
- Close All Unnecessary Foreground Applications

Windows NT Tuning Overview

The Windows NT Server operating system offers considerably fewer tuning adjustments than UNIX. This difference constrains systems administrators when they try to optimize Windows NT Server performance, but it also makes Windows NT Server easier to use.

There are still ways, however, to make Windows NT Server a better application server environment for Oracle9*i* database. Most of the operating system specific procedures described in this chapter enable Oracle9*i* database to reserve more system resources, such as CPU, memory, and disk I/O.

In addition, because Oracle9*i* database is a high-performance database management system that effectively uses resources of your Windows NT computer, it should not also serve as any of the following:

- Primary or backup domain controller
- File or print server
- Remote access server
- Router

These configurations consume network, memory, and CPU resources. In addition, the Windows NT computer that is running your Oracle9*i* database should not be locally accessed with a high frequency or intensively used for local user processing, unless it has enough resources to accommodate all this activity.

Note: Information presented in this chapter does not apply to Oracle9*i* database installations on Windows NT Workstation. It does apply to Oracle9*i* database installations on Windows NT Server and Windows NT Server 4.0, Enterprise Edition. Windows NT Server 4.0, Enterprise Edition includes capability for using additional products. One such product is Microsoft Cluster Server, which is required to implement Oracle Fail Safe (OFS).

Reduce Priority of Foreground Applications on Server Console

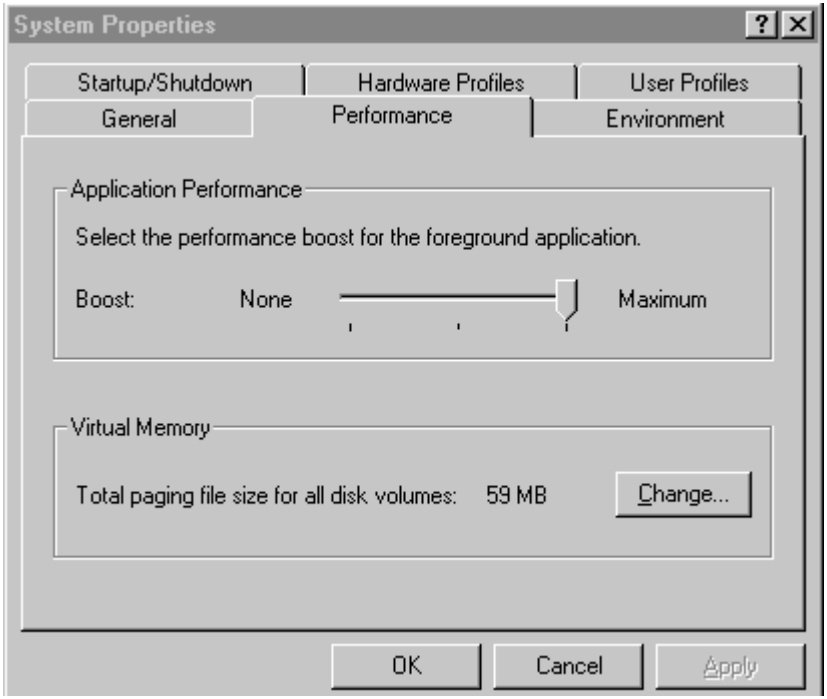
One of the settings provided by default during installation of Windows NT Server gives interactive foreground applications priority over every background **process**. To prevent foreground applications on the server console from taking excessive processor time away from Oracle9*i* database, you can reduce priority for foreground applications:

1. Choose Start > Settings > Control Panel > System.

   The System Properties dialog box appears.

2. Choose the Performance tab.

   The Performance tab displays:



3. Move the Application Performance Boost slider to None.
4. Click OK.
5. Exit the Control Panel.

Configure Windows NT Server to Be an Application Server

Windows NT memory manager divides up system memory into three different pools described in Table 5–1.

Table 5–1 Windows NT Server Memory Shares

Pool	Percent of Total Memory
Kernel and other system services	9%
File Cache	41%
Paged Memory	50%

Windows NT Server memory manager tries to balance each application's usage of memory by dynamically paging memory between physical RAM and a virtual memory paging file. If an application is particularly memory-intensive (like Oracle9*i* database) or if a large number of applications are running concurrently, then combined memory requirements of the applications may exceed physical memory capacity.

The large proportion of memory reserved for file caching (41%) can be quite beneficial to file and print servers. But it may not be advantageous to application servers that often run memory-intensive network applications. A Windows NT Server file cache is particularly unnecessary for Oracle9*i* database, which performs its own caching through **System Global Area (SGA)**.

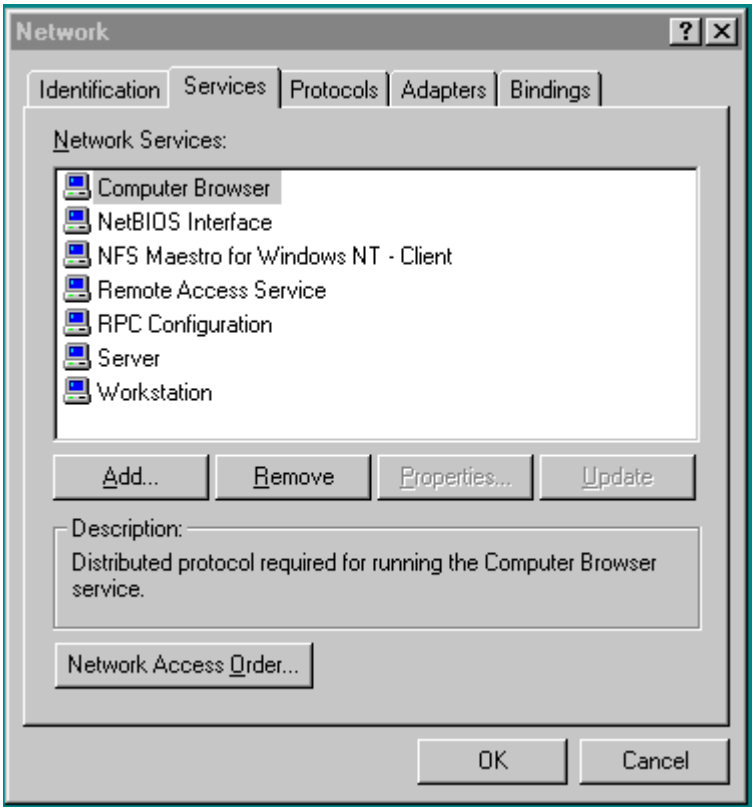
You can reset the Windows NT Server memory model from the default file and print server, with its large file cache, to a network applications model, with a reduced file cache and more physical memory available for Oracle9*i* database.

To configure Windows NT Server to be an applications server:

1. Choose Start > Settings > Control Panel > Network.

   The Network dialog box appears.

2. Choose the Services tab.



3. Select the Server service and choose Properties.
4. Select the radio button for a network applications configuration and choose OK.
5. Click OK on the Network dialog box.
6. Exit the Control Panel.
7. Restart the system for changes to take effect.

Disable Unnecessary Services

After you have significantly reduced the file cache as described in "Configure Windows NT Server to Be an Application Server" on page 5-4, you can retrieve additional physical memory for Oracle9*i* database by disabling services not needed for core operating system functionality. Unnecessary services include:

- `License Logging Service`
- `Plug and Play`
- `Remote Access Autodial Manager`
- `Remote Access Connection Manager`
- `Remote Access Server`
- `Telephony Service`

Consult with your systems administrator to identify other unnecessary services.

Do *not* disable any of the following services:

- `Alerter`
- `Computer Browser`
- `EventLog`
- `Messenger`
- `OracleService`*`SID`*
- `Oracle`*`HOME_NAME`*`TNSListener`
- `OracleStart`*`SID`* (Oracle8 release 8.0.6 and earlier)
- `Remote Procedure Call (RPC) Service`
- `Server`
- `Spooler`
- `TCP/IP NetBIOS Helper`
- `Workstation`

To disable unnecessary services:

1. Choose Start > Settings > Control Panel > Services.

   The Services dialog box appears.

2. Select the service to disable.
3. Choose Startup.

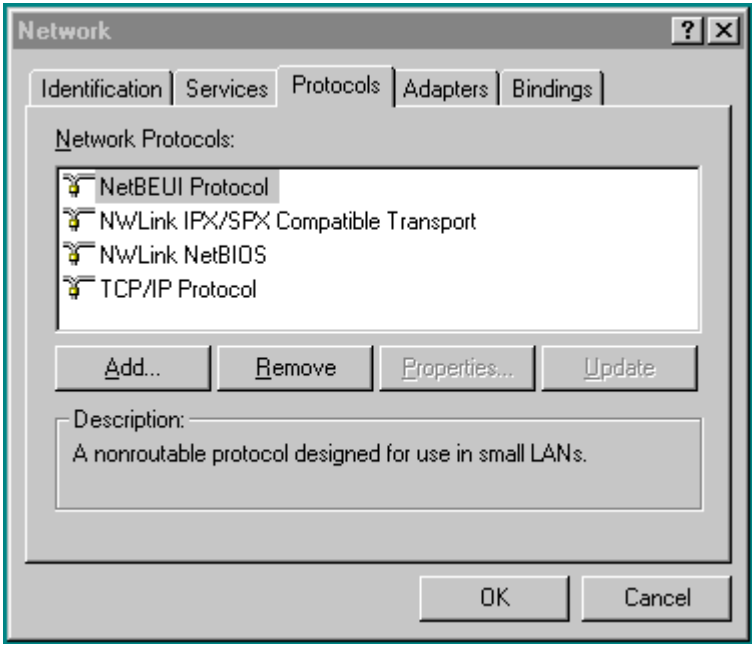
   The Service dialog box appears.
4. Select Disabled in the Startup Type field.
5. Click OK.
6. Exit the Control Panel.

Remove Unused Network Protocols

Remove all unnecessary network protocols on Windows NT Server so that processing time can be concentrated on servicing only critical protocols:

1. Choose Start > Settings > Control Panel > Network.

   The Network dialog box appears.
2. Choose the Protocols tab.



3. Identify which protocols are necessary for server functionality. Oracle Corporation recommends you use TCP/IP as the sole network protocol.
4. Select unneeded network protocols and choose Remove.
5. Click Yes in the confirmation dialog box.
6. Repeats Steps 4 and 5 until only necessary protocols remain.
7. Exit the Control Panel.

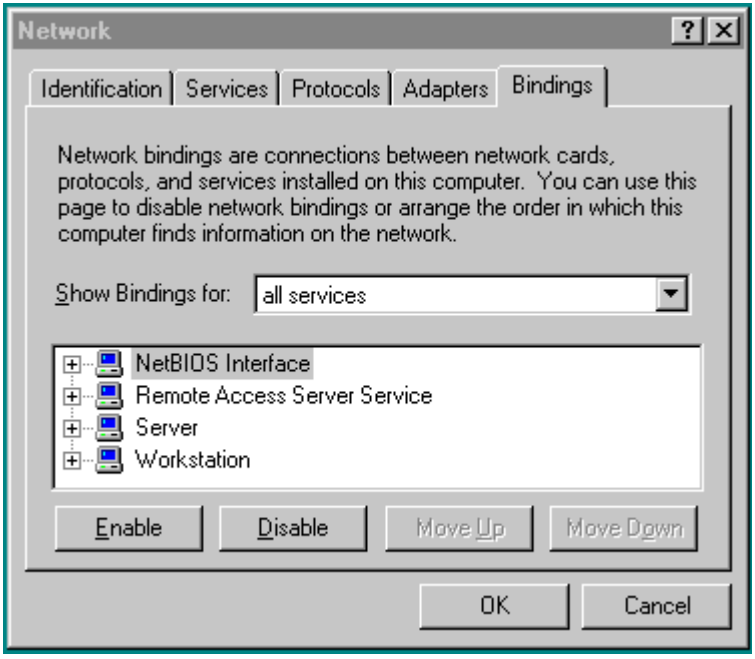
Reset Network Protocol Bind Order

If multiple protocols must be installed on the server, you can give the protocol most frequently used by Oracle9*i* database highest priority by resetting the network protocol bind order:

1. Choose Start > Settings > Control Panel > Network.

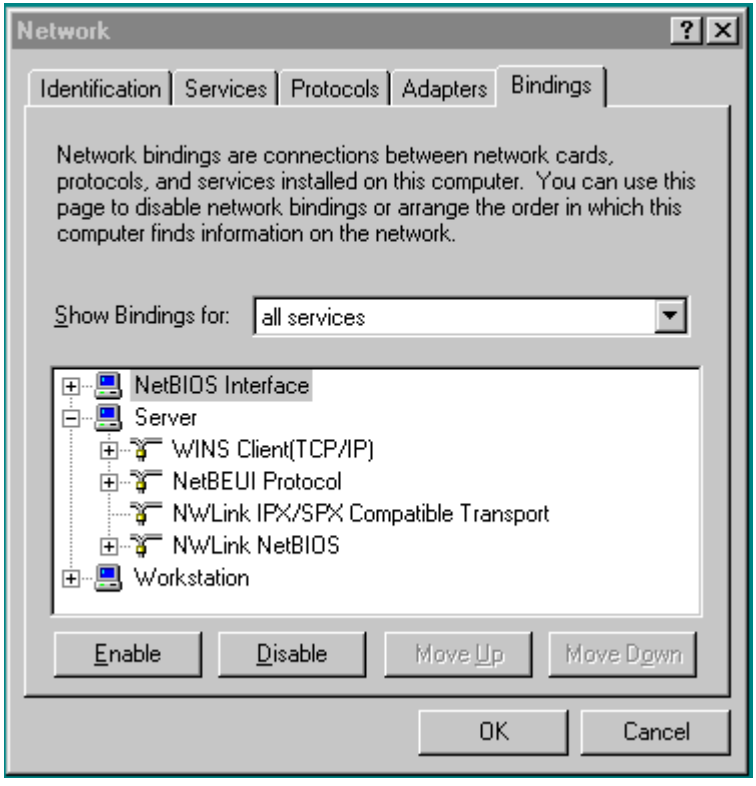
   The Network dialog box appears.

2. Choose the Bindings tab. The Bindings tab displays:



3. Select all services from the Show Bindings list box.
4. Double-click Server to expand the list of currently installed services.

   All available protocol adapters display:



5. If the primary Oracle9*i* database protocol is not at the top of the list, select it.
6. Click Move Up until the primary protocol is at the top of the list.
7. If multiple network interface cards (NICs) are installed, expand each protocol and move the NIC used most often by Oracle9*i* database to the top of the list.
8. Click OK to save the changes.
9. Exit the Control Panel.
10. Restart Windows NT Server to enable the new changes.

Apply Latest Reliable Windows NT Server Service Pack

Microsoft releases operating system patches, called Service Packs, on a quarterly basis. Service Packs are sometimes abbreviated as SP*x* where *x* is the release number of the Service Pack. For example, Windows NT Server 4.0 SP4 denotes a Windows NT Server 4.0 installation with Service Pack 4 applied. Service Packs are a collection of bug fixes and product enhancements to the basic Windows NT Server release. In general, apply Service Packs as soon as it is safe to do so, since they fix bugs and can improve Windows NT Server performance or functionality.

While Service Packs are supposed to fix bugs, there have been reports (for example, the initial release of Windows NT Server 4.0 SP2) of bugs within the patch updates themselves. In general, it is safest to wait a few weeks after a Service Pack is released before implementing it. This allows time for other field sites to report any problems with the SP release.

The latest Windows NT Server Service Packs may be downloaded as self-extracting archives from:

```
http://support.microsoft.com/support/ntserver/content/servicepacks/default.asp
```

To install a Service Pack:

1. Download the Service Pack.
2. Read the README file, which contains important installation instructions.

   Unless there are assurances that the Service Pack works without flaws on Windows NT Server, choose to create an Uninstall directory. This enables the Service Pack to be removed and the original configuration to be restored.

   Service Pack files overwrite similarly-named files in the previous Windows NT Server configuration. However, Service Pack files can be overwritten in turn by setup programs that copy files from the original installation media.

   For example, installing a new network protocol or printer driver usually requires copying files from original Windows NT Server installation media. When Service Pack files are comprehensively or selectively overwritten, the Service Pack must be re-applied.

To uninstall a Service Pack (if an Uninstall directory was created):

1. Launch program `UPDATE.EXE` (or self-extracting archive).
2. Follow appropriate prompts.

Use Hardware or Operating System Striping

Compared to CPU and memory speeds, hard disk drives are extremely slow. Now that hard disk drives are relatively inexpensive, Oracle Corporation recommends that Windows NT Server use logical volumes comprised of striped physical disks. Data striping is an effective means of reducing the impact of relatively slow hard drives by distributing file I/O across a number of hard drives simultaneously.

Striping data across a number of disks is one example of a redundant array of inexpensive disks (RAID). There are several different types of RAID, also referred to as RAID levels, ranging from high performance to high reliability. The three most common RAID levels in Oracle9*i* database installations are RAID-0, RAID-1, and RAID-5. Descriptions of each RAID level follow Table 5–2, which shows each level's read and write penalties.

Table 5–2 RAID Levels in Oracle9i Database Installations

RAID Level	Read Penalty[1]	Write Penalty[2]
0 (Disk Striping)	1:1	1:1
1 (Disk Mirroring	1:1	2:1
0 + 1	1:1	2:1
5 (Distributed Data Gathering)	1:1	4:1

[1] Read penalty is ratio of I/O operations to read requests.

[2] Write penalty is ratio of I/O operations to write requests.

Disk Striping

RAID level 0 enables high-performance, nonfault tolerant disk striping. Multiple physical hard disks are aggregated into a logical whole, either by a disk controller or through the operating system (for example, Windows NT Server stripe sets). Data operations against the logical volume are broken down into as many chunks as there are physical drives in the array, making simultaneous use of all disks. Given identical hard disks, if one hard disk has a throughput rate of DISKRATE operations/second, then a RAID 0 logical volume has a rate of:

```
(DISKRATE * [number of physical  drives in array]) operations/second
```

The downside of RAID 0 is its lack of fault tolerance. If one disk in the logical volume fails, the whole volume fails and must be restored from a **backup**.

Disk Mirroring

RAID level 1 enables fault tolerant disk mirroring with some chance of a performance penalty. Essentially, every write to a mirrored disk is duplicated on another drive dedicated to this purpose (the mirror drive). If the mirrored disk fails, the mirror drive is brought online in real time. After the faulted drive is replaced, the mirror configuration can be reestablished.

The read penalty for RAID level 1 is nominally 1:1, but it may benefit from split reads on some controllers. When the controller knows which mirror can be accessed fastest, for example, it can lower seek times by directing I/O operations to that disk.

Disk Striping + Mirroring

RAID level 0+1 enables mirroring of an array of striped hard disks. This is a blend of RAID 0 and RAID 1, offering high-performing fault tolerance.

Distributed Data Guarding

RAID level 5, also known as disk striping with parity, eliminates the costly requirement to mirror. In RAID 5, multiple hard disks are aggregated into a striped logical volume, similar to RAID 0, but each drive contains parity information such that any single drive failure is tolerated. With one failed drive, a RAID-5 system can allow continued access to data, although access times are greatly reduced due to on-the-fly rebuilding of bytes from parity information. RAID-5 solutions usually allow hot-swapping of faulty drives with replacements, triggering a rebuild of the failed drive's data onto the replacement from parity information.

The write penalty of 4:1 results from 2 reads and 2 writes during parity calculation.

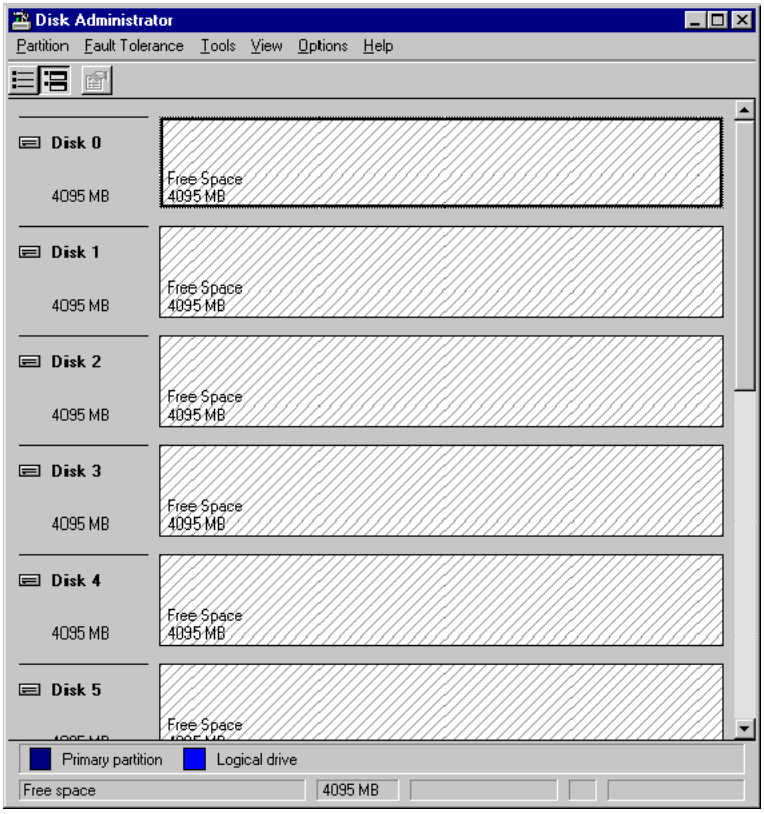
Windows NT Server Stripe Set Example

This example demonstrates creation of a Windows NT Server stripe set. Suppose that there are six SCSI-2 hard disks attached to a common non-RAID disk controller. Each hard drive is not yet partitioned.

To create a Windows NT Server stripe set:

1. Choose Start > Programs > Administrative Tools (Common) > Disk Administrator.

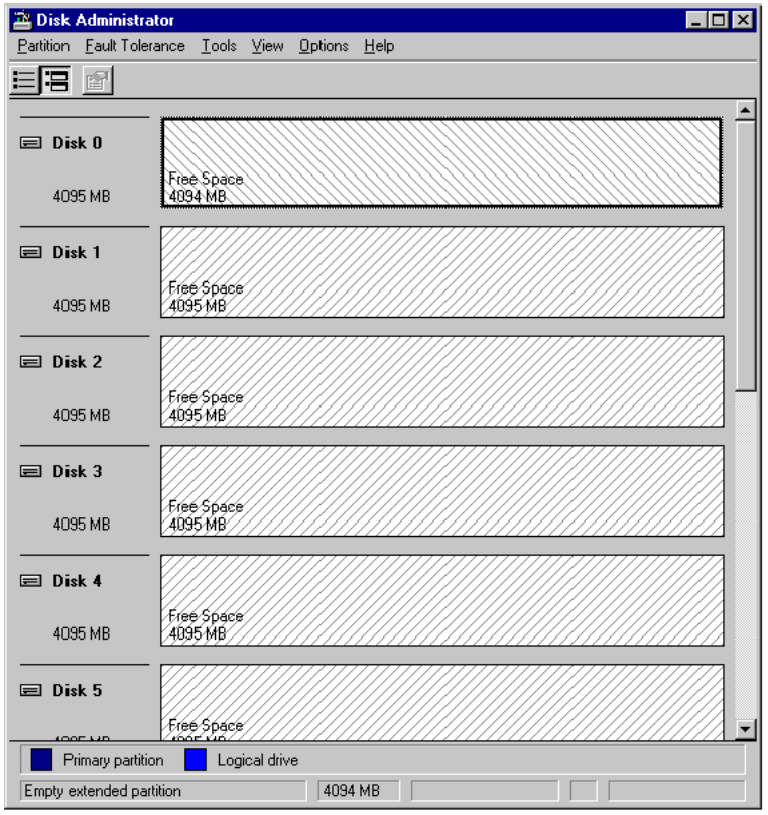
   The Disk Administrator dialog box appears.



2. Choose drive 0's free space.

3. Choose Create Extended from the Partition menu to create an extended partition sized to the hard disk's total capacity.

   Note that the hash marks filling partitioned drive 0's free space are now in the opposite direction to marks in the free space of the unpartitioned hard disks:



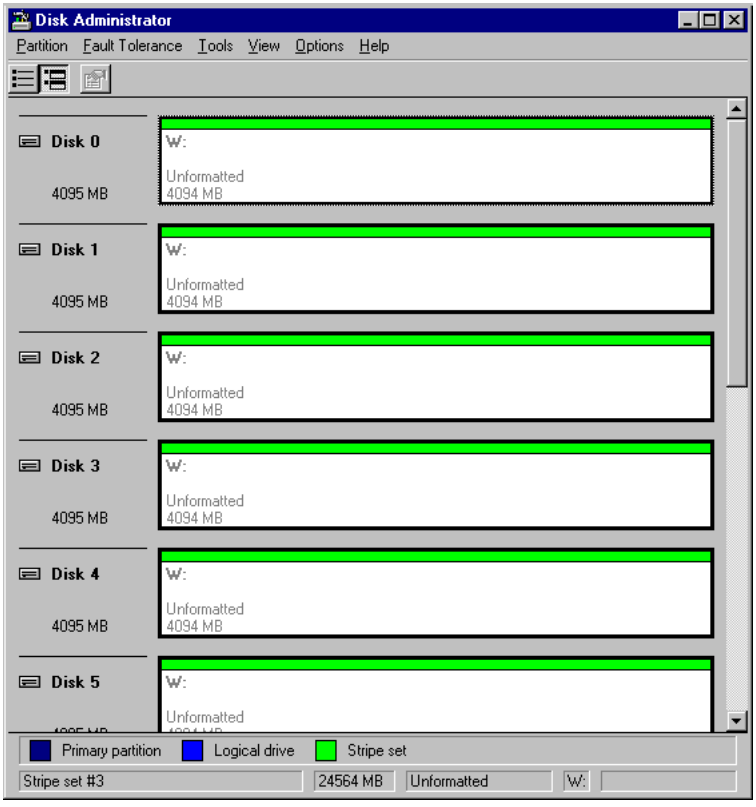
4. Repeat Steps 2 and 3 for each of the five remaining unpartitioned hard disks.
5. Choose the first (topmost) drive.
6. Press Ctrl while choosing the remaining five drives so that all six hard disks are selected.
7. Choose Create Stripe Set from the Partition menu.
8. Click OK on the Logical Volume Size dialog box. The maximum total size of the stripe set volume will be:

   ```
   [number of hard disks] * [capacity of the smallest selected partition]
   ```

   Disk Administrator automatically adjusts sizes of each drive's stripe if a stripe set size of less than the maximum allowable is chosen.

9. Choose Commit Changes Now from the Partition menu to save volume changes.

The logical stripe set volume across all physical hard disks appears.



10. Restart the system to enact the change. After restarting, the new logical stripe set volume is ready to be formatted.

Note: It is not necessary to create an extended partition on each hard disk before creating a stripe set. However, it is useful to do so for cases when a drive's stripe does not occupy its entire capacity and additional non-stripe set logical volumes are created on the drive. Extended partitions are necessary, however, when creating more than four logical partitions due to a Windows NT limit of only four primary (nonextended) partitions.

Multiple Striped Volumes for Sequential and Random Access

If there are enough physical disks in Windows NT Server, create at least two striped volumes (in addition to a standalone hard disk or striped volume for the operating system). One striped volume can be used for sequential data access, and the other can be used for random data access.

Oracle9*i* database redo logs and archived redo logs, for example, are written in sequential order. Because of reduced head movement, hard disks perform best when reading or writing sequential data.

Oracle9*i* database data files, however, are usually accessed in random order. Random access in a hard disk results in significant head movement, translating to slower data access.

Unless redo logs are separated from datafiles (at physical device level), undo file I/O contention may result, increasing access times for both types of files.

Multiplex Windows NT Server Virtual Memory Paging File

Some virtual memory paging is likely even if Oracle9*i* database is the only network application running on Windows NT Server, because Windows NT Server memory manager will attempt to move seldom-used pages to disk to free up more physical memory for hot pages.

Multiplexing the Windows NT Server virtual memory paging file is a good strategy to boost overall system performance. Splitting the paging file onto at least two different physical volumes (or logical volumes as long as underlying physical volumes do not overlap) provides a significant performance boost to virtual memory swapping operations.

Even though this is a good technique to increase speed of virtual memory paging, too much paging activity is still a performance hit and should be corrected by adding more RAM to the server.

General Page File Sizing Tip

Oracle Corporation recommends that total combined size of your page files be at least equal to physical RAM on your computer. Configurations where combined size is two to four times the size of physical RAM are not uncommon. Minimize paging as much as possible. But situations in which the operating system runs out of or low on paging space are to be avoided at all costs. Adequately-sized paging files spaced across physical disks spread out I/O most efficiently, because the operating system spreads paging evenly across page files.

Note: Internal read/write batch size for Windows is 4K.

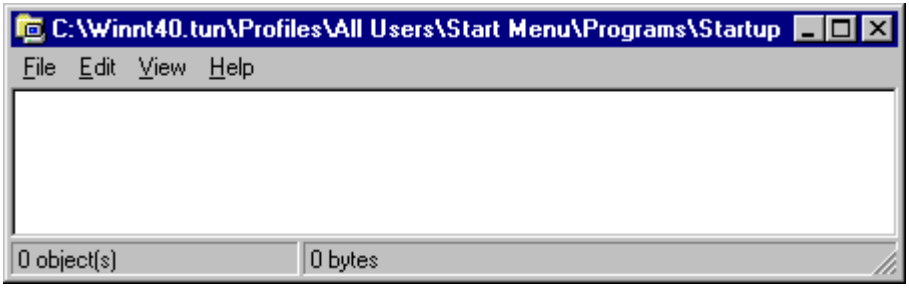
Close All Unnecessary Foreground Applications

Once procedures in previous sections have been applied, remember to close any unnecessary foreground applications. Three are described in this section:

- Startup Folder
- Virtual DOS Machines
- Screen Savers

Startup Folder

Remove applications from Startup folders of Windows NT Server console operators.



Virtual DOS Machines

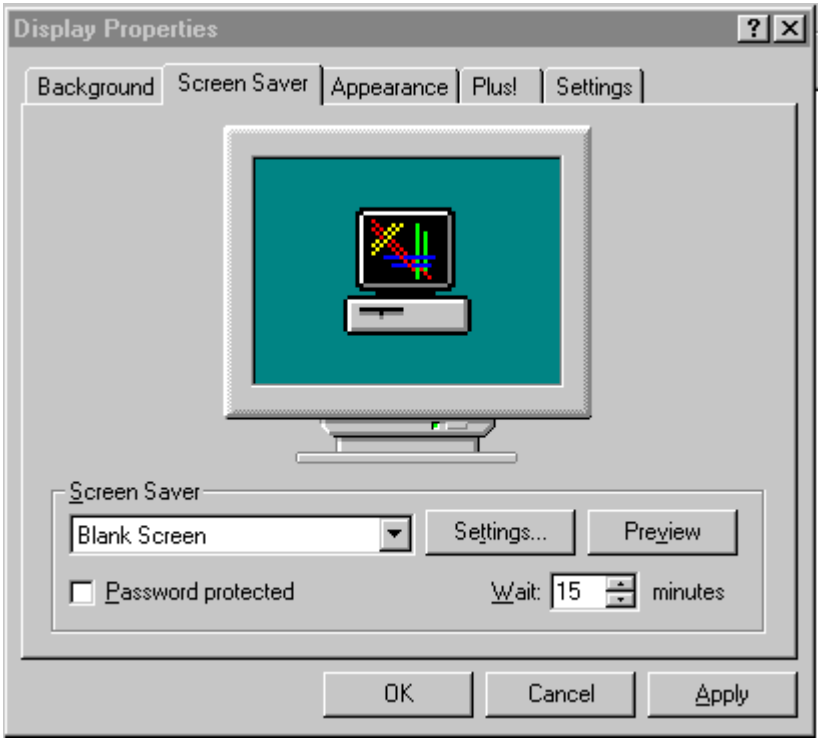
Database administrators frequently run SQL scripts from command prompts (also known as virtual DOS machines, or VDMs). While VDMs are relatively lightweight, screen painting VDMs is expensive. When executing long-running scripts from a VDM, minimize the window so that the system can focus on the operation and not on a flood of window repaint messages.

Screen Savers

Most entertaining screen savers quickly saturate the CPU. If a screen saver must be run, choose Blank Screen, which uses the least amount of processing time. To set a blank screen saver:

1. Choose Start > Settings > Control Panel > Display.

   The Display Properties dialog box appears.



2. Choose the Screen Saver tab.
3. Select Blank Screen from the Screen Saver list.
4. Click OK.
5. Exit the Control Panel.

6

Backing Up and Recovering Database Files

This chapter provides information on backing up and recovering your database.

This chapter contains these topics:

- Selecting a Backup and Recovery Tool
- Backing Up and Recovering Files with OCOPY Utility

Selecting a Backup and Recovery Tool

This section describes tools available for **backup** and **recovery** of your Oracle9*i* database and makes recommendations on which tool to use.

This section contains these topics:

- Oracle Enterprise Manager
- Recovery Manager in Command-line Mode
- Third-party Vendor Products
- OCOPY Utility

Note: Before performing a backup when Oracle9*i* database is shut down, and regardless of the backup tool you use, stop service `OracleService`*`SID`* to clear locks on database files. If you do not stop this service, some database files may not be backed up.

Oracle Enterprise Manager

Oracle Enterprise Manager DBA Management Pack includes Backup and Recovery Wizards for backing up and recovering an Oracle9*i* database. These wizards also provide a graphical user interface (GUI) to Recovery Manager.

Backup Wizard enables users to:

- Perform database backups to disk and tape
- Create backup scripts
- Specify frequency, time, backup configuration, and multiple destinations
- Back up an entire database or **tablespaces**, data files, and archived redo logs
- Perform backups both online and when the database is shut down

Oracle highly recommends Oracle Enterprise Manager for backing up and recovering your Oracle9*i* database because of its capabilities and ease of use.

Note: Backing up to tape requires an optional media management layer (MML) from a third-party vendor.

> **See Also:** *Oracle Enterprise Manager Administrator's Guide* for more information on Oracle Enterprise Manager

Recovery Manager in Command-line Mode

Recovery Manager in command-line mode is an Oracle9*i* tool that manages backing up, restoring, and recovering files. Recovery Manager is automatically installed with Oracle Utilities. Recovery Manager uses a special **PL/SQL** interface to the server for invoking backup functions. This interface operates invisibly; you will interface only with a command line.

> **Note:** On Windows NT, starting with Oracle8*i* release 8.1.6, additional transformations are added to file names entered by the user. If you use Recovery Manager, existing file names stored in the recovery catalog must be normalized. To normalize file names, see *Oracle9i Database Migration*.

When RMAN needs to resynchronize from a read-consistent version of the control file, it creates a temporary snapshot control file. The default location of this file on Windows NT is *ORACLE_BASE**ORACLE_HOME*\database\sncf*SID*.ora.

You must clearly understand the command-line syntax described in *Oracle9i Recovery Manager Reference* before using this tool. Backing up to tape requires an optional MML from a third-party vendor.

> **See Also:** "Database Tools Overview" in *Oracle9i Database Getting Started for Windows* for instructions on accessing this tool and *Oracle9i Recovery Manager User's Guide* for instructions on using this tool

Third-party Vendor Products

Third-party vendors provide tape backup software that works with Backup Wizard and Recovery Manager. Contact your third-party vendors for additional information.

OCOPY Utility

OCOPY utility is a command-line tool that can make the following types of backups:

- Backups when Oracle9*i* database is shut down
- Backups of a tablespace that is offline when the database is open
- Backups of a tablespace that is online when the database is open

OCOPY can back up only to disk and can perform only local backups.

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

Integrating RMAN with a Media Manager on Windows NT

On Windows NT, Oracle accesses the media management library through library `orasbt.dll`. This file must exist in the system path. Typically, the file is located in Oracle home folder *`ORACLE_HOME`*`\bin`.

Note: You are not required to start or shut down the **instance** when installing the media management library.

If an `orasbt.dll` already exists in the system path, then remove it before installing the media manager. After installation, check your media management vendor documentation to determine where the media management library is installed.

Note: File `orasbt.dll` is not required to be in folder *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\bin` as long as the folder containing the library is in the system PATH variable setting. To see the PATH variable setting, choose Start > Settings > Control Panel > System > Environment.

Backing Up and Recovering Files with OCOPY Utility

Use OCOPY utility for both hot and raw database file backups of the following file types:

- File Allocation Table (FAT) files
- NT File System (NTFS) files
- Logical raw files
- Physical raw files

Logical raw files are stored on a logical drive (accessed with direct I/O) identified by `\\.\`*x* where *x* is logical drive designator. Physical raw files are stored on a physical hard drive (accessed with direct I/O) identified by `\\.\physicaldrive`*n* where *n* is 0, 1, 2, and so on, representing each of the physical drives in the system. For more information on raw partitions, see Appendix A, "Storing Tablespaces on Raw Partitions".

Copy Modes

OCOPY utility has two copy modes:

- Hard disk mode copies hot files to a new name and location on a hard disk, where you can use an archive utility to back them up.
- Multiple diskettes mode backs up large hot files directly to multiple diskettes.

Note: If the file to be copied is too large to fit on one diskette, OCOPY utility prompts you to insert new diskettes as needed. OCOPY utility catalogs parts of the file automatically so that it can be reconstructed at a later time.

OCOPY Backup Syntax

Table 6–1 describes syntax to use when backing up all file types to hard disk. Table 6–2 does the same for backups to multiple diskettes. Table 6–3 defines syntax terms.

Table 6–1 OCOPY Backup Syntax in Hard Disk Mode

File Type	Syntax
FAT or NTFS files	`C:\> ocopy old_file new_file`
Logical raw	`C:\> ocopy \\.\x: new_file`
Physical raw	`C:\> ocopy \\.\physicaldrive n new_file`

Table 6–2 OCOPY Backup Syntax in Multiple Diskettes Mode

File Type	Syntax
FAT or NTFS files	`C:\> ocopy /B old_file a:`
Logical raw	`C:\> ocopy /B \\.\x: a:`
Physical raw	`C:\> ocopy /B \\.\physicaldrive n a:`

Table 6–3 OCOPY Backup Syntax Definitions

Term	Definition
old_file	Name and location of hot file to back up
new_file	Name and location of backup copy
x	Raw drive that holds a single raw database file
physicaldrive*n*	Physical drive that holds a single raw database file
/B	Large files must be split over multiple diskettes
a:	Drive containing diskette(s) on which to save backup copy.

Note: Always use fresh diskettes for OCOPY utility procedures. Do not back up a file onto a diskette that contains part or all of another file backed up using OCOPY.

OCOPY Recover Syntax

Table 6–4 describes syntax to use when recovering all file types from hard disk. These are files originally backed up without option /B. Table 6–5 describes syntax to use when recovering all file types from multiple diskettes. These are files originally backed up with option /B. Table 6–6 defines syntax terms.

Note: When recovering from multiple diskettes, insert the diskette containing the first part of the backed up file before starting OCOPY utility.

Table 6–4 OCOPY Recover Syntax in Hard Disk Mode

File Type	Syntax
FAT or NTFS files	Use Windows copy command
Logical raw	C:\> ocopy *new_file* \\.*x*:
Physical raw	C:\> ocopy *new_file* \\.\physicaldrive*n*

Table 6–5 OCOPY Recover Syntax in Multiple Diskettes Mode

File Type	Syntax
FAT or NTFS files	C:\> ocopy /R a: *restore_dir*
Logical raw	C:\> ocopy /R a: \\.*x*:
Physical raw	C:\> ocopy /R a: \\.\physicaldrive*n*

Table 6–6 OCOPY Recover Syntax Definitions

Term	Definition
new_file	Name and location of backup copy
x	Raw drive that holds a single raw database file
physicaldrive*n*	Physical drive that holds a single raw database file
/R	Restore option
a:	Drive containing diskette with backed up file
restore_dir	Server directory in which to place file (Restored file has same name as original file.)

7

Oracle9*i* Database Specifications for Windows

Oracle9*i* for Windows uses initialization parameters to enable various features of the database every time an **instance** is started.

This chapter contains these topics:

- Initialization Parameter File Overview
- Sample File
- SGA_MAX_SIZE Parameter
- Initialization Parameters Without Windows NT-Specific Values
- Displaying Initialization Parameter Values
- Uneditable Database Initialization Parameters
- Calculating Database Limits

Initialization Parameter File Overview

An **initialization parameter file** is an ASCII text file containing parameters. By changing parameters and values in an initialization file, you can specify, for example:

- Amount of memory Oracle9*i* database uses
- Whether to archive filled online redo logs
- Which **control files** currently exist

Every database instance has a corresponding initialization parameter file and `ORACLE_SID` **registry** parameter that points to the **System Identifier (SID)** for the instance.

The initialization parameter filename takes the form `init.ora`. A single instance might have several initialization parameter files, each having some differences that affect system performance.

Note: Your `init.ora` file for initialization parameters is set by Oracle Universal Installer during database installation. These parameter settings may vary depending on your hardware configuration. For descriptions of all initialization parameters and instructions for setting and displaying their values, see *Oracle9i Database Reference*.

Location of Initialization Parameter File

By default, Oracle9*i* uses initialization parameter files located in

```
ORACLE_HOME\admin\db_name\pfile\init.ora
```

unless you specify a different initialization file with option `PFILE` at database startup.

Note: If you create a database manually using a SQL script, you are required to create an initialization parameter file or copy an existing initialization parameter file and modify the contents. If you use Database Configuration Assistant to create a database, the initialization parameter file is automatically created for you.

Editing Initialization Parameter File

To customize Oracle9*i* database functions, you may be required to edit the initialization parameter file. Use only an ASCII text editor to modify the file.

Sample File

A sample file called `initsmpl.ora` is located in directory

*ORACLE_BASE**ORACLE_HOME*\admin\sample\pfile.

To use file `initsmpl.ora` as part of database creation:

1. Rename the file to `init.ora`.
2. Edit this file to reflect the correct location of your database control files and the name of your database, as a minimum.

If you installed a **starter database**, the initialization parameter file used by the starter database is located in

*ORACLE_BASE**ORACLE_HOME*\admin\sample\pfile

You can use either `initsmpl.ora` or the starter database `init.ora` as a basis for creating a new Oracle9*i* database initialization parameter file.

The annotated, sample initialization parameter file contains alternative values for initialization parameters. These values and annotations are preceded by comment signs (#), which prevent them from being processed. To activate a particular parameter, remove the preceding # sign. To de-activate a particular parameter, edit the initialization parameter file to add a comment sign.

For example, several initialization parameters are specified with three different values to create a small, medium, or large **System Global Area (SGA)**, respectively. The parameter that creates a small SGA is active in the following example:

```
  db_block_buffers = 200  # SMALL
# db_block_buffers = 550  # MEDIUM
# db_block_buffers = 3200 # LARGE
```

To create a medium-sized SGA, comment out the small parameter definition and activate the medium parameter definition. Edit the initialization parameter file as follows:

```
# db_block_buffers = 200  # SMALL
  db_block_buffers = 550  # MEDIUM
# db_block_buffers = 3200 # LARGE
```

SGA_MAX_SIZE Parameter

Parameter SGA_MAX_SIZE holds the maximum size that System Global Area (SGA) can reach for a particular instance. Beginning with release 1 (9.0.1), Oracle9*i* can change its SGA configuration while the instance is running. This allows sizes of buffer cache, shared pool, and large pool to be changed without instance shutdown.

Oracle9*i* can start instances underconfigured and allow the instance to use more memory by growing SGA up to a maximum of SGA_MAX_SIZE. If no SGA_MAX_SIZE value is specified, then Oracle selects a default value that is the sum of all components specified or defaulted at initialization time. If SGA_MAX_SIZE specified in the initialization parameter file is less than the sum of all components specified or defaulted to at initialization time, then the setting of SGA_MAX_SIZE in the initialization parameter file is ignored.

See Also:

- *Oracle9i Database Performance Guide and Reference* for more information about SGA initialization parameters.
- *Oracle9i Database Concepts* for more information about SGA and its components

Memory allocated for an instance's SGA is displayed on instance startup when using Oracle Enterprise Manager (or SQL*Plus). You can also display the current instance's SGA size by using the SQL*Plus SHOW statement with the SGA clause.

Initialization Parameters Without Windows NT-Specific Values

Oracle9i Database Reference describes default values for many initialization parameters as being operating system-specific. However, not all parameters that *Oracle9i Database Reference* describes as having operating system-specific values affect Windows NT. In these cases, Windows NT uses either default value set in the Oracle9*i* kernel or does not use the parameter. Table 7–1 describes these initialization parameters:

Table 7–1 Initialization Parameters Without Windows NT-Specific Values

Parameter	Description
AUDIT_FILE_DEST	Not supported on Windows NT and should not be added to the initialization parameter file
DB_WRITER_PROCESSES	Supported, but typically unnecessary due to Windows NT asynchronous I/O capabilities
COMPATIBLE_NO_RECOVERY	Uses default value set in Oracle9*i* kernel (no Windows NT-specific value)
BACKGROUND_CORE_DUMP	Specifies whether Oracle includes SGA in core file for Oracle background processes
SHADOW_CORE_DUMP	Specifies whether Oracle includes SGA in core file for foreground (client) processes
CORE_DUMP_DEST	Specifies directory where Oracle dumps core files
CPU_COUNT	Oracle9*i* automatically sets value to number of processors available for your Oracle instance
HI_SHARED_MEMORY_ADDRESS	Not applicable to Windows NT
SHARED_MEMORY_ADDRESS	Not applicable to Windows NT
LARGE_POOL_SIZE	Uses maximum value limited by available memory
LOG_BUFFER	Starter database uses value set in Oracle9*i* kernel (no Windows NT-specific value). The Custom database creation option of Database Configuration Assistant enables you to customize the value for this parameter.
ORACLE_TRACE_COLLECTION_PATH	Uses default value set in Oracle9*i* kernel (no Windows NT-specific value)
ORACLE_TRACE_FACILITY_NAME	Uses default value set in Oracle9*i* kernel (no Windows NT-specific value)
ORACLE_TRACE_FACILITY_PATH	Uses default value set in Oracle9*i* kernel (no Windows NT-specific value)
SPIN_COUNT	Uses default value set in Oracle9*i* kernel (no Windows NT-specific value)

Displaying Initialization Parameter Values

Windows NT-specific parameter values can be viewed by using an ASCII editor to open the initialization parameter file:

```
ORACLE_BASE\ORACLE_HOME\admin\db_name\pfile\init.ora
```

To display any parameter value (whether set in the initialization parameter file or the Oracle9*i* kernel), enter the following command at the SQL*Plus command prompt:

```
SQL> SHOW PARAMETER parameter_name
```

where *parameter_name* is the name of a specific initialization parameter.

Uneditable Database Initialization Parameters

Check the following initialization parameters in Table 7–2 when creating a new database. They cannot be modified after you have created the database. See Chapter 1, "Postinstallation Database Creation" for details on creating a new database, including the part of the procedure when you modify these parameters.

Table 7–2 Uneditable Database Initialization Parameters

Parameter	Description
CHARACTER SET[1]	Specifies database **Globalization Support** character set to use. This parameter can be set only when you create the database.
DB_BLOCK_SIZE	Specifies size in bytes of standard Oracle database blocks.
DB_NAME	Specifies name of the database to be created. Database name is a string of eight characters or less. You cannot change the name of a database.

[1] Not an initialization parameter, but rather a clause in the CREATE DATABASE statement. See Chapter 1, "Postinstallation Database Creation" for an example of using this clause.

Calculating Database Limits

Use size guidelines in this section to calculate Oracle9*i* database limits.

Table 7–3 Block Size Guidelines

Type	Size
Maximum block size	16,384 bytes or 16 kilobytes (KB)
Minimum block size	2 kilobytes (KB)
Maximum blocks for each file	4,194,304 blocks
Maximum possible file size with 16 K sized blocks	64 Gigabytes (GB) (4,194,304 * 16,384) = 64 gigabytes (GB)

Table 7–4 Maximum Number of Files for Each Database

Block Size	Number of Files
2 KB	20,000
4 KB	40,000
8 KB	65,536
16 KB	65,536

Table 7–5 Maximum File Sizes

Type	Size
Maximum file size for a FAT file	4 GB
Maximum file size in NTFS	16 Exabytes (EB)
Maximum database size	65,536 * 64 GB equals approximately 4 Petabytes (PB)
Maximum control file size	20,000 blocks

Table 7–6 Maximum Number of Extents for Each Database

Typical Block Sizes	Number of Extents
2 KB	121
4 KB	255
8 KB	504
16 KB	1032
32 KB	2070

Table 7–7 Shadow Process Memory

Release Number	Memory
Oracle9*i* release 2 (9.2)	444 KB
Oracle9*i* release 1 (9.0.1)	444 KB
Oracle8*i* release 8.1.7	335 KB
Oracle8*i* release 8.1.6	335 KB
Oracle8 release 8.1.3	265 KB
Oracle8 release 8.0.5	254 KB
Oracle8 release 8.0.4	254 KB

A

Storing Tablespaces on Raw Partitions

This appendix describes how to configure your system to store datafiles for **tablespaces** on **raw partitions**.

This appendix contains these topics:

- Raw Partition Overview
- Creating an Extended Partition
- Creating a Tablespace in a Raw Partition

> **Note:** Oracle Real Application Clusters require additional configuration tools. See *Oracle9i Real Application Clusters Setup and Configuration* for information on creating logical partitions and assigning symbolic links. Do not use this appendix to create partitions for Oracle Real Application Clusters.

Raw Partition Overview

In addition to storing datafiles for tablespaces on a file system, datafiles can also be stored on raw partitions.

A raw partition is a portion of a physical disk that is accessed at the lowest possible level. Input/output (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 after generation of an extended partition and a logical partition. Windows NT Disk Administrator enables you to create an extended partition on a physical drive.

Note: The Windows 2000 equivalent is Disk Management.

An extended partition points to raw space on the disk that can be assigned multiple logical partitions for database files. An extended partition also avoids the four-partition limit by letting you define large numbers of logical partitions to accommodate applications using the Oracle9*i* database server. Logical partitions can then be given symbolic link names to free up drive letters.

Raw partitions are of two types:

- Physical Disk
- Logical Partition

Note: Although you can use physical disks, Oracle Corporation recommends that you use logical partitions.

Physical Disk

A physical disk represents the entire disk and points to the following:

`\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 Disk Administrator and matching the *x* in `\Device\Harddiskx\Partition0`. `\\.` Symbolic link name `\PhysicalDrive`*x* is automatically defined by Windows for every hard disk in the computer. For example, a computer with three hard disks:

```
\\.\PhysicalDrive0
\\.\PhysicalDrive1
\\.\PhysicalDrive2
```

Internally, these names expand to the following:

```
\\.\PhysicalDrive0 = \Device\Harddisk0\Partition0
\\.\PhysicalDrive1 =\Device\Harddisk1\Partition0
\\.\PhysicalDrive2 =\Device\Harddisk2\Partition0
```

`Partition0` is special, because it represents the entire physical disk regardless of any partitioning scheme on that disk. On all disks recognized by Windows, Disk Administrator writes a signature on the first block of all disks. To avoid overwriting that block, Oracle9*i* database skips the first block of a physical raw partition that is used for an Oracle9*i* datafile.

Logical Partition

A logical partition is a partition created by 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 `\\.\ACCOUNTING_1`. 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 can expand to:

```
\\.\D:= \Device\Harddisk2\Partition1
\\.\ACCOUNTING_1= \Device\Harddisk3\Partition2
```

Drive letters can be assigned to specific partitions, using Disk Administrator. Symbolic link names can, on the other hand, be assigned using a utility such as `DOSDEV.EXE`, which is available with Windows NT Resource Kit.

Note: Oracle9*i* database does not skip the first block of a logical raw partition used for an Oracle9*i* datafile.

Physical Disk and Logical Partition Considerations

Consider the following when deciding which raw partition to use:

- Physical disks are automatically defined by Windows to represent the entire disk, and should never be defined by the user.
- Logical partitions must 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 Oracle9*i* datafile and using a partition that occupies the entire disk for an Oracle9*i* datafile are 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 Oracle9*i* datafile (`Partition0`), use the pre-defined physical raw names that 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 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:

```
\\.\PhysicalDriveACCOUNTING_1 = \Device\Harddisk2\Partition1
\\.\PhysicalDriveACCOUNTING_2 = \Device\Harddisk3\Partition1
```

Answer: Oracle9*i* database handles datafiles 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 to use physical disk naming conventions. Oracle Corporation recommends that you convert to the logical partition at your earliest convenience. See "Compatibility Issues" on page A-5.

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

```
\\.\ACCOUNTING_1 = \Device\Harddisk1\Partition0
```

Answer: This poses severe problems, because Disk Administrator typically writes a signature into the first block of every disk, and consequently may overwrite a portion of the datafile header.

This can also cause data loss. Never use `Partition0` with the logical partition convention. See "Compatibility Issues" on page A-5 for information on rebuilding your Oracle9*i* 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 OCOPY utility to copy data to/from a raw partition for both physical and logical raw conventions. See "Compatibility Issues" on page A-5 for further information.

Compatibility Issues

Physical and logical partition conventions are not compatible with one another because of the extra block that is skipped for physical raw conventions. This also means you cannot simply use OCOPY utility to copy from a physical disk to a logical partition, because 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. Recreate the database by 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 preceding steps.

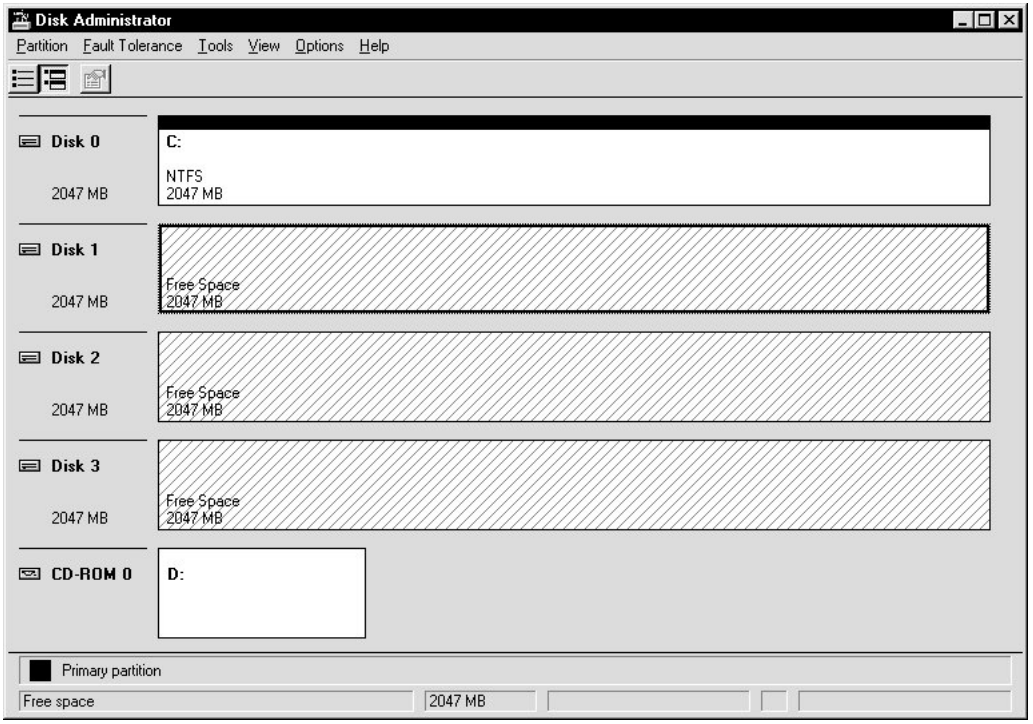
Creating an Extended Partition

Only one extended partition can be created for each disk. You can use 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. Choose Start > Programs > Administrative Tools > Disk Administrator.

   The Disk Administrator window appears.



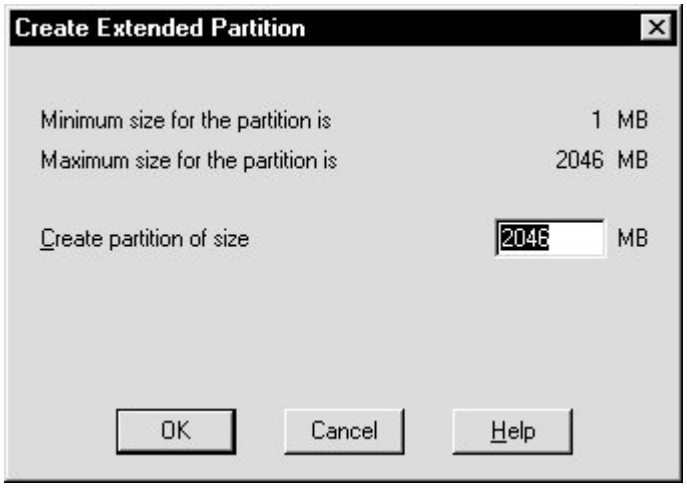
Note that the lines display diagonally from top right to bottom left, indicating unpartitioned devices.

2. 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 Corporation recommends that you use the entire disk.

3. Choose Partition > Create Extended.

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



4. Use the default maximum size, then choose OK.

> **Note:** Changes that you have made are not saved until you choose Partition > Commit Changes Now or exit Disk Administrator.

The extended partition is created.

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

Creating 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.

> **Note:** Oracle Corporation recommends that you not create more than 120 logical partitions in an extended partition.

To create logical partitions in an extended partition:

1. Select an area of free space in an extended partition by clicking the mouse on it.
2. Choose Partition > Create.

Disk Administrator displays minimum and maximum sizes for the logical partition.

3. Enter the size of the logical partition for the data file, then choose OK.

   The size depends on how large your datafiles will be. Add 2 MB to this size for overhead.

4. Repeat Steps 1-3 for each additional datafile that you plan to store in a raw partition.

5. Choose Partition > Commit Changes Now.

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

6. Click Yes.

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

7. Click OK.

8. Write down the hard disk number(s) and the number of the partition (starting at 1) for that drive. Oracle Corporation recommends using a worksheet similar to the one shown here.

Hard Disk Number	Partition Number Range
Hard Disk*x*	Partitions *x*-*x*
Hard Disk*x*	Partitions *x*-*x*

9. Choose Partition > Close.

   Disk Administrator exits.

Creating a Tablespace in a Raw Partition

To create a tablespace using a datafile located in a raw partition:

1. Start SQL*Plus:

   ```
   C:\> sqlplus
   ```

2. Connect to the Oracle9*i* repository database:

   ```
   Enter user-name: SYSTEM/password
   ```

   where *password* for user account **SYSTEM** is MANAGER by default. If you have changed this password, substitute the correct password for MANAGER.

3. Create the tablespace:

   ```
   SQL> CREATE TABLESPACE tablespace DATAFILE '\\.\datafile' SIZE xm;
   ```

   where:

   - *tablespace* is the tablespace name
   - '\\.\' is the drive letter or symbolic link name assigned to the raw partition
   - *x* is the tablespace size in megabytes (Twenty megabytes is a good starting place.)

For example, to create a tablespace named accounting_1 that was assigned a symbolic link name of accounting_1, enter the following:

```
SQL> CREATE TABLESPACE accounting_1 DATAFILE '\\.\accounting_1 SIZE 502M;
```

Note: If you are creating a database with the SQL script, modify datafiles that are stored on raw partitions with a naming convention of \\.\drive_letter: or \\.\symbolic link name.

Glossary

authentication

Identification of a user, device, or other entity in a computer system, often as a prerequisite for allowing access to resources in a system.

backup

A representative copy of data. This copy includes important parts of your database such as **control files**, redo log files, and data files.

A backup is a safeguard against unexpected data loss; if you lose your original data, you can use the backup to make data available again. A backup is also a safeguard against an application error; if an application makes incorrect changes, you can restore the backup.

control files

Files that record the physical structure of a database and contain database name, names and locations of associated databases and online **redo log files**, timestamp of database creation, current log sequence number, and checkpoint information.

data dictionary

A set of read-only tables that provide information about a database.

downgrade

To convert data in an Oracle database into an earlier release. See **upgrade** and **migrate**.

external routine

A function written in a third-generation language (3GL), such as C, and callable from within **PL/SQL** or SQL as if it were a PL/SQL function or procedure.

Globalization Support

Oracle architecture that ensures that database utilities, error messages, sort order, date, time, monetary, numeric, and calendar conventions automatically adapt to native language and locale.

HOME*ID*

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.

HOME_NAME

Represents the name of an ***ORACLE_HOME***. In Oracle 8*i* release 8.1.6 or later, all Oracle homes have a unique *HOME_NAME*.

initialization parameter file

An ASCII text file that contains information needed to initialize a database and **instance**.

instance

Every running Oracle database is associated with an Oracle instance. When a database is started on a database server (regardless of computer type), Oracle allocates a memory area called **System Global Area (SGA)** and starts at least one Oracle **process**. This combination of SGA and one or more Oracle processes is called an instance. Memory and processes of an instance manage the associated database's data efficiently and serve one or more database users.

instantiate

Produce a more defined version of some object by replacing variables with values (or other variables).

listener

Server **process** that listens for and accepts incoming connection requests from client applications. Oracle listener processes start up Oracle database processes to handle subsequent client communications.

listener.ora

A configuration file that describes one or more Transparent Network Substrate (TNS) listeners on a server.

migrate

To **upgrade** or **downgrade** an Oracle database or convert data in a non-Oracle database into an Oracle database.

mount

To associate a database with an **instance** that has been started.

multiple Oracle homes

Capability of having more than one Oracle home on a computer.

net service name

Name used by clients to identify **Oracle Net** server. A net service name is mapped to a port number and protocol. Also known as a connect string, database alias, or service name.

network listener

A listener on a server that listens for connection requests for one or more databases on one or more protocols. See **listener**.

Optimal Flexible Architecture (OFA)

A set of file naming and placement guidelines for Oracle software and databases.

ORACLE_HOME

Corresponds to the environment in which Oracle products run. This environment includes location of installed product files, PATH variable pointing to products' binary files, **registry** entries, **net service name**, and program groups.

If you install an OFA-compliant database, using Oracle Universal Installer defaults, Oracle home (known as *\ORACLE_HOME* in this guide) is located beneath *X:\ORACLE_BASE*. It contains subdirectories for Oracle software executables and network files.

Oracle Net

Oracle network interface that enables Oracle tools running on network workstations and servers to access, modify, share, and store data on other servers.

Oracle service

A **service** that is associated with an Oracle component.

ORACLE_BASE

Oracle base, known as *ORACLE_BASE* in this guide, is the root of the Oracle directory tree.

If you install an OFA-compliant database using Oracle Universal Installer defaults, *ORACLE_BASE* is *X*:\oracle where *X* is any hard drive (for example, C:\oracle).

PL/SQL

Oracle Corporation's procedural language extension to SQL.

PL/SQL enables you to mix SQL statements with procedural constructs. You can define and execute PL/SQL program units such as procedures, functions, and packages.

privilege

A right to execute a particular type of SQL statement or to access another user's object.

process

A mechanism in an operating system that can run an executable. (Some operating systems use the terms job or task.) A process normally has its own private memory area in which it runs. On Windows NT a process is created when a program runs (such as Oracle or Microsoft Word). In addition to an executable program, all processes consist of at least one **thread**. The ORACLE master process contains hundreds of threads.

raw partitions

Portions of a physical disk that are accessed at the lowest possible disk (block) level.

recovery

To restore a physical **backup** is to reconstruct it and make it available to the Oracle server. To recover a restored backup is to update it using redo records (that is, records of changes made to the database after the backup was taken). Recovering a backup involves two distinct operations: rolling forward the backup to a more current time by applying redo data, and rolling back all changes made in uncommitted transactions to their original state.

redo log files

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

redo log buffer

A circular buffer in System Global Area (SGA) that contains information about changes made to the database.

registry

A Windows repository that stores configuration information for a computer.

remote computer

A computer on a network other than the local computer.

remote database

A database on a computer other than the local database.

replication

Copying and maintaining database objects in multiple databases that make up a distributed database system.

role

A named group of related privileges. You can grant a role to users or other roles.

schema

A named collection of objects, such as tables, **views**, clusters, procedures, and packages, associated with a particular user.

service

An executable **process** installed in the Windows **registry** and administered by Windows. Once a service is created and started, it can run even when no user is logged on to the computer.

SID

See **System Identifier (SID)**.

snapshot

(1) Information stored in rollback segments to provide transaction recovery and read consistency. Rollback segment information can be used to re-create a snapshot of a row before an update.

(2) A read-only copy of a master table located on a remote node. Snapshots can be queried, but not updated; only the master table can be updated. Snapshots are periodically refreshed to reflect changes made to the master table.

starter database

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

SYSDBA

A special database administration **role** that contains every system **privilege** with ADMIN OPTION, and **SYSOPER** system privilege. SYSDBA also permits CREATE DATABASE actions and time-based recovery.

SYSOPER

A special database administration **role** that permits a database administrator to perform STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG, and RECOVER, and includes RESTRICTED SESSION **privilege**.

System Global Area (SGA)

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

System Identifier (SID)

A unique name for an Oracle **instance**. To switch between Oracle databases, users must specify the 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 a **listener.ora** file.

SYSTEM

One of two standard DBA usernames automatically created with each database. (The other username is SYS.) SYSTEM is created with an initial password of MANAGER. Username SYSTEM is the preferred username for DBAs to use for database maintenance.

tablespaces

A database is divided into one or more logical storage units called tablespaces. Tablespaces are divided into logical units of storage called segments, which are further divided into extents.

thread

An individual path of execution within a **process**. Threads are objects within a process that execute program instructions. Threads allow concurrent operations within a process so that a process can execute different parts of its program simultaneously on different processors. A thread is the most fundamental component that can be scheduled on Windows NT.

tnsnames.ora

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

trace file

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

upgrade

To convert data in an Oracle database into a later release. See **downgrade** and **migrate**.

username

A name that can connect to and access objects in a database.

views

Selective presentations of the structure of, and data in, one or more tables (or other views).

Index

Symbols

\\.\PhysicalDrive*x*, A-2

A

Advanced Replication support
  data dictionary tables, 2-14
  initialization parameter requirements, 2-13
  tablespace requirements, 2-13, 2-14
alert files, 4-13
  monitoring a database, 4-2
  using, 4-13
ALTER DATABASE ARCHIVELOG
    command, 3-21
ARCHIVELOG mode, 3-21
archiving mode
  controlling, 3-20
  custom database, 3-20
  starter database, 3-20
archiving procedures, 3-22
  for redo log files, 3-20
  specifying an archive destination, 3-22
  UNC not allowed, 3-20
audit trail
  managing, 4-12
AUDIT_FILE_DEST, 4-13
AUDIT_FILE_DEST parameter, 7-5
authentication
  using a password file, 3-15
auto-starting
  Oracle Services, 3-7

B

BACKGROUND_DUMP_DEST, 4-13
backup
  databases, 1-26
  OCOPY file types, 6-5
  stopping services to perform cold backups, 6-4
backup and recovery tools
  selecting, 6-2
block size
  maximum, 7-7
  parameter, 7-7
blocks for each file, maximum, 7-7

C

cold backups, performing, 6-4
commands
  ALTER DATABASE ARCHIVELOG, 3-21
  ARCHIVE LOG LIST, 3-21
  CREATE DATABASE, 1-21
  NET START, 1-22
  REGEDT32, 1-24
configuring
  Oracle Spatial, 2-8
  Oracle Text, 2-5
CONNECT / AS SYSDBA
  using, 3-10, 3-21
connecting
  to a database, 3-10, 3-21
control files
  additional backslashes in r. 8.0.4, 1-3, 3-20
  maximum size, 7-7
CREATE DATABASE command, 1-21

custom database
archiving mode, 3-20

D

data dictionary
Advanced Replication tables, 2-14
data warehousing
importing a sample schema, 1-14
Database Configuration Assistant
deleting databases, 1-15
database files
deleting, 1-18
database monitoring
using Performance Monitor, 4-2
with alert files, 4-13
with Event Viewer, 4-9
with trace files, 4-13
databases
backing up, 1-26
connecting to, 3-10, 3-21
creating manually, 1-15
deleting, 1-18
exporting, 1-17
importing, 1-23
monitoring, 4-2
naming conventions, 1-2
password encryption, 3-19
shutting down, 3-9, 3-11
starting, 3-9
DBLINK_ENCRYPT_LOGIN, 3-19

E

encrypting, database passwords, 3-19
error messages
ORA-00256, 1-3, 3-20
ORA-01102, 1-2
ORA-09291, 1-3, 3-20
OSD-04018, 1-3, 3-20
Event Viewer, 4-2
blue (informational) icon, 4-11
for monitoring a database, 4-2
managing, 4-12
reading, 4-11
red (stop sign) icon, 4-11
using, 4-9
yellow (exclamation point) icon, 4-11
Export Utility
databases, 1-17
interactive mode, 1-17
parameter mode, 1-17
extended partition
creating, A-5
defined, A-2
extents
maximum number for each database, 7-7

F

files
alert, 4-13
maximum number for each database, 7-7
maximum size possible, 7-7
sample init.ora, 7-3
trace, 4-13

H

Hostname parameter, 4-3

I

Import Utility
databases, 1-23
interactive mode, 1-23
parameter mode, 1-23
index size, calculating, 7-7
initialization parameter file
defined, 7-2
editing, 7-3
location, 7-2
initialization parameter requirements
Advanced Replication Support, 2-13
instances
modifying, 1-31
running multiple instances, 3-15
interactive mode
Export Utility, 1-17
Import Utility, 1-23

L

LOG_ARCHIVE_DEST_n, 1-3, 3-20, 3-22
LOG_ARCHIVE_FORMAT parameter, 3-22
LOG_ARCHIVE_START, 3-22

M

mapped drives, 1-3, 3-20
MAXDATAFILES, 1-20
memory usage, 5-4
monitoring
  alert files, 4-2
  Event Viewer, 4-2
  Management Pack, 4-2
  Performance Monitor, 4-2
  trace files, 4-2
multiple instances, running, 3-15
multiple Oracle homes
  service naming conventions, 3-2

N

naming conventions for multiple Oracle homes, 3-2
NET START command, 1-22
NOARCHIVELOG mode, 3-21

O

OCOPY
  backup file types, 6-5
OLTP
  importing a sample schema, 1-14
ORA_ENCRYPT_LOGIN, 3-19
ORA-00256 error, 1-3, 3-20
Oracle Enterprise Manager
  Management Pack, 4-2
Oracle Managed Files, 1-21
Oracle Services
  auto-starting, 3-7
  naming conventions for multiple Oracle homes, 3-2
  shutting down a database, 3-11
  starting, 3-5
  stopping, 3-6
Oracle Spatial
  configuring, 2-8
Oracle Text
  configuring, 2-5
ORACLE_*SID*, 1-24, 3-15
Oracle9*i* database
  connecting to, 3-10, 3-21
  password encryption, 3-19
  selecting a backup and recovery tool, 6-2
  shutting down, 3-9, 3-11
  specifications, 7-7
  starting, 3-9
ORADEBUG
  debugging utility, 3-24
  starting, 3-24
ORADIM
  modifying an instance, 1-31
  moving or copying password files, 3-17
ORAPWD
  creating password files, 3-15

P

parameter mode
  Export Utility, 1-17
  Import Utility, 1-23
parameters
  AUDIT_FILE_DEST, 4-13
  BACKGROUND_DUMP_DEST, 4-13
  block size, 7-7
  DBLINK_ENCRYPT_LOGIN, 3-19
  Hostname, 4-3
  LOG_ARCHIVE_DEST_n, 1-3, 3-20, 3-22
  LOG_ARCHIVE_FORMAT, 3-22
  LOG_ARCHIVE_START, 3-22
  ORA_ENCRYPT_LOGIN, 3-19
  ORACLE_*SID*, 1-24, 3-15
  Password, 4-3
  REMOTE_LOGIN_PASSWORDFILE, 3-16
  USER_DUMP_DEST, 4-13
  Username, 4-3
partitions
  extended, A-2
  logical partition, A-3
  physical disk, A-2

raw, A-2
password
encryption, 3-19
utility, 3-15
password file
authenticating database administrators, 3-15
creating, 3-15
hiding, 3-16
viewing, 3-17
Password parameter, 4-3
Performance Monitor, 4-2
accessing, 4-4
for monitoring a database, 4-2
Hostname parameter, 4-3
Password parameter, 4-3
registry information, 4-3
troubleshooting information, 4-9
Username parameter, 4-3
using, 4-2
Performance Pack, monitoring a database, 4-2
PhysicalDrive, A-2
PWD*SID*.ORA file, 3-15

R

raw partition
defined, A-2
logical partition, A-3
overview, A-2
physical disk, A-2
recovering
Oracle9*i* database, 6-2
redo log files
archiving, 3-20
REGEDT32 command, 1-24
registry
Hostname, 4-3
Password, 4-3
update ORACLE_SID, 1-24
Username, 4-3
remote computers, accessing database files, 3-20
REMOTE_LOGIN_PASSWORDFILE, 3-16

S

schemas
data warehousing, 1-14
OLTP, 1-14
services, 3-2
auto-starting, 3-7
shutting down a database, 3-11
starting, 1-20, 3-5
stopping, 3-6
SET ORACLE_SID=SID, 3-15
shared server
configuring, 2-9
disabling, 2-9
enabling, 2-9
shutting down
databases, 3-9, 3-11
specifying an archiving file format, 3-22
SQL*Plus
shutting down the database, 3-9
starting, 3-9
starting the database, 3-9
using ORADEBUG, 3-24
starter database
archiving mode, 3-20
starting
databases, 3-9
Oracle Services, 3-5
Performance Monitor, 4-4
SQL*Plus, 3-9
stopping, Oracle Services, 3-6
symbolic links
creating, A-9

T

tablespaces
Advanced Replication support, 2-13, 2-14
trace files, 4-13
creating with ORADEBUG, 3-24
for monitoring a database, 4-2
USER_DUMP_DEST, 4-13
using, 4-13
troubleshooting
with ORADEBUG, 3-24

tuning Windows NT Server operating system, 5-2

U

UNC, 1-3, 3-20
Universal Naming Convention, 1-3, 3-20
USER_DUMP_DEST, 4-13
Username parameter, 4-3

V

viewing
  password file, 3-17

W

Windows NT tuning
  applying latest service packs, 5-10
  closing unnecessary foreground applications, 5-17
  configuring server to be an application server, 5-4
  disable unnecessary services, 5-6
  multiple striped volumes for sequential and random access, 5-16
  overview, 5-2
  reducing priority of foreground applications on server, 5-3
  removing unused network protocols, 5-7
  resetting the network protocol bind order, 5-8
  screen savers, 5-18
  startup folder, 5-17
  using hardware and operating system striping, 5-11
  virtual DOS machines, 5-18
Windows NT-specific
  archiving procedures, 3-22
  audit trail availability, 4-11
  audit trail facility, 4-11
  automatic database startup after restart, 3-7
  initialization parameter file, 7-2
  parameter file location, 7-2
  parameter filename and location, 7-2
  password filename and location, 3-15
  trace file names, 4-13