Oracle9*i*

Database Administrator's Guide

Release 1 (9.0.1) for Windows

June 2001

Part No. A90164-01

ORACLE

Oracle9i Database Administrator's Guide, Release 1 (9.0.1) for Windows

Part No. A90164-01

Primary Author: Herbert Kelly III, Mark Kennedy, and Helen Slattery.

Contributors: Server Technologies Division - NT Development and Robert Farrington of Oracle Support Services.

Contents

2 Postinstallation Configuration Tasks

3 Administering a Database

4 Monitoring a Database

5 Tuning Windows NT to Optimize Oracle9*i*

6 Backing Up and Recovering Database Files

7 Oracle9*i* Database Specifications for Windows

A Storing Tablespaces on Raw Partitions

Glossary

Index

Send Us Your Comments

Oracle9*i* Database Administrator's Guide, Release 1 (9.0.1) for Windows

Part No. A90164-01

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:

- E-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. Contact information for Oracle Support Services is available at this Web site:

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

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 preface contains these topics:

Audience

This guide is necessary for anyone installing, configuring, or administering Oracle9*i* Enterprise Edition, Oracle9*i* Standard Edition or Oracle9*i* Personal Edition.

Note: This guide describes *only* the features of Oracle9*i* Enterprise Edition and Oracle9*i* software that apply to the Windows NT, Windows 95, Windows 98, and Windows 2000 operating systems. For information about these products that is applicable to *all* operating systems, see the other documentation included in your package. Instructions for installing the documentation library and viewing its contents are in three README files at the root level of the documentation CD-ROMs:

Organization

This guide is organized as follows:

Chapter 1, "Postinstallation Database Creation"

This chapter describes how to create a database after installation manually or with the Oracle Database Configuration Assistant.

Chapter 2, "Postinstallation Configuration Tasks"

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

Chapter 3, "Administering a Database"

This chapter describes how to administer the Oracle9*i* database.

Chapter 4, "Monitoring a Database"

This chapter describes how to monitor the Oracle9*i* database.

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 recommendations and procedures for backing up and recovering database files.

Chapter 7, "Oracle9i Database Specifications for Windows"

This chapter describes Oracle9*i* initialization parameters on Windows NT and Windows 2000 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 create raw partitions for database tablespaces.

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

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

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

Conventions

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

- Conventions in Text
- Conventions in Code Examples
- Conventions 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.	The C datatypes such as **ub4**, **sword**, or **OCINumber** are valid. When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles, emphasis, syntax clauses, or placeholders.	*Oracle9i Database Concepts* You can specify the *parallel_clause*. Run `U`*`old_release`*`.SQL` where *old_release* refers to the release you installed prior to upgrading.

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 using the `BACKUP` command. Query the `TABLE_NAME` column in the `USER_TABLES` data dictionary view. Specify the `ROLLBACK_SEGMENTS` parameter. Use the `DBMS_STATS.GENERATE_STATS` procedure.
lowercase monospace (fixed-width font)	Lowercase monospace typeface indicates executables 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.	Enter `sqlplus` to open SQL*Plus. 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.

Conventions in Code Examples

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

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

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

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

Convention	Meaning	Example
`...`	Horizontal ellipsis points indicate either: ■ That we have omitted parts of the code that are not directly related to the example ■ 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.	
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 variables for which you must supply particular values.	`CONNECT SYSTEM/system_password`

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

lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.	`SELECT last_name, employee_id FROM employees;` `sqlplus hr/hr`

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 Oracle Database Configuration Assistant, you must click the Start button on the taskbar and then choose Programs > Oracle - *HOME_NAME* > Database Administration > Database Configuration Assistant.	Choose Start > Programs > Oracle - *HOME_NAME* > Database Administration > Database Configuration Assistant

Convention	Meaning	Example
File and Directory Names	File / 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 manual.	`C:\oracle\oradata>`

	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.	`Oracle`*`HOME_NAME`*`TNSListener`
	The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	

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:\orawin95` for Windows 95 ■ `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 Oracle9*i* Release 1 (9.0.1) on a computer with no other Oracle software installed, the default setting for the first Oracle home directory is `C:\oracle\ora90`. 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 compliances and for information on installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

Documentation Accessibility

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

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

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

What's New in Oracle9*i* for Windows

This section describes new features of Oracle9*i* release 1 (9.0.1). Information on desupported and deprecated features is also included.

The following sections describe the new features in Oracle9*i* for Windows operating systems:

Windows Integration

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

Development and Deployment Improvements

Windows security supports Oracle Wallets in the registry, in Active Directory, and 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.

Oracle Database Configuration Assistant Improvements

The Oracle 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 the Oracle 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 the 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.

See Also: *Oracle9i Database Getting Started for Windows*

Oracle9*i* Desupported and Deprecated Features

CONNECT INTERNAL

`CONNECT INTERNAL` and `CONNECT INTERNAL/PASSWORD` are not supported in Oracle9*i*. Instead, the following are the exact equivalents:

```
CONNECT / AS SYSDBA
```

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

Server Manager

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 manually or by using the Oracle Database Configuration Assistant after installing Oracle.

This chapter contains these topics:

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. A name is associated with a database at creation time and stored in its control files. If the database keyword is provided in the `CREATE DATABASE` statement or when prompted by the Oracle Database Configuration Assistant, that value becomes the name for that database.

If you attempt to **mount** two Oracle9*i* databases with the same database name, you receive the following error during the 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, 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 file**(s) and specify a new database name. This restriction only exists for Oracle8*i* and later releases. Any Oracle7 instances running simultaneously with an Oracle9*i* **instance** are not subject to this restriction.

Note: The 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, your directory paths will differ.

Accessing Data Files and Log Files on Remote Computers

Although it is possible for Oracle to access database files on **remote computer**s using the Universal Naming Convention (UNC), it is not recommended because of performance and network reliability concerns.

UNC is a PC format for specifying the location of resources on a local area network. UNC uses the following format:

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

For example, to access the file `system01.dbf` in the directory `C:\oracle\oradata\orcl` on the shared server `argon`, you reference the file as:

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

Note that the location of archive log files cannot be specified using UNC. If you set the `LOG_ARCHIVE_DEST_`*n* initialization parameter to a UNC specification, the 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
```

Ensure that you set the `LOG_ARCHIVE_DEST_`*n* initialization parameter to a mapped drive.

Note: An ORA-00256 error also occurs if you enter:

```
\\\meldell\rmdrive
```

or

```
\\\meldell\\rmdrive
```

Control files required the additional backslashes for release 8.0.4, but **redo log files** and datafiles did not.

Creating a Database Using Tools

You can choose either of the following methods to create a database:

- Using Oracle Database Configuration Assistant
- Creating a Database Manually

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

If you prefer you can also create a database using command line tools.

Using Oracle Database Configuration Assistant

Use the 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, make sure `everyone` is a member of Pre `Windows 2000 Compatible` domain group. If `everyone` is not a member of this group, 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, the database on the member server will randomly get ACCESS DENIED errors when accessing Active Directory.

The database service on the server has to be running as a LocalSystem or domain user, for Database Configuration Assistant to successfully register the database with Active Directory. If the database is running as a local user, then trying to register the database with Active Directory using Database Configuration Assistant fails, as this user cannot logon to Active Directory. After successfully registering with the directory using the Database Configuration Assistant, if the database service is running as a LocalSystem, then manually add the computer name to the Access Control List of the OracleDBSecurity container (in Active Directory) with read permissions on OracleDBSecurity container. If the database service is running as a domain user, then the username should be manually added to the Access Control List of the OracleDBSecurity container (in Active Directory) with read permissions on OracleDBSecurity Container. If this is not done, then you may not be able to use the Active Directory to get enterprise **role**s.

Oracle 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 Oracle Database Configuration Assistant in standalone mode (that is, after installation). It can also be used during the installation procedure to create a database.

Create a Database

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

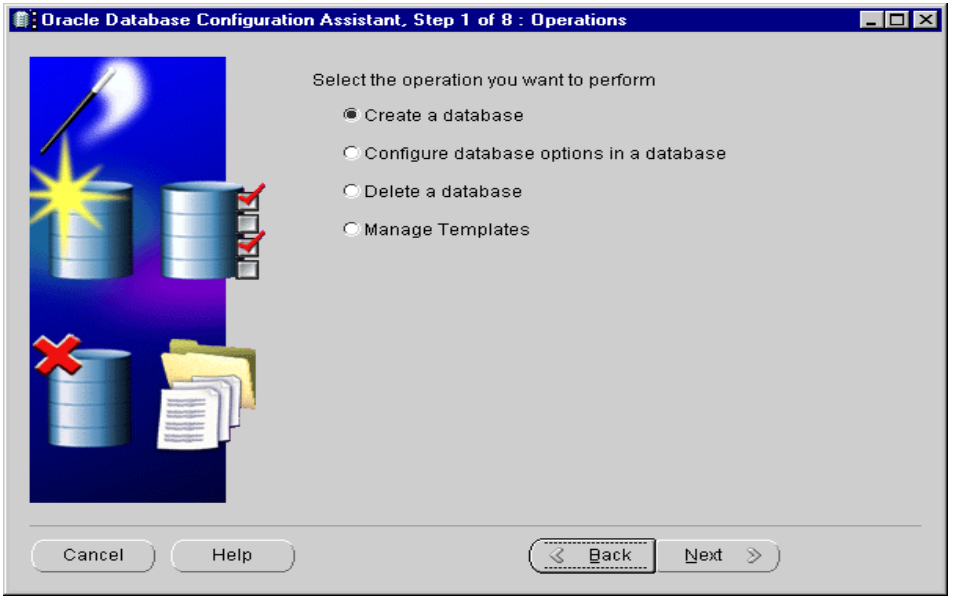
To create a database using Oracle Database Configuration Assistant:

> **Note:** Users must have Windows NT Administrator's **privilege**s in order to create an Oracle9*i* database. If Oracle Database Configuration Assistant is run from a user account that is not part of the Administrator's group, it displays a warning stating that you do not have administrative privileges to create the database. Log in as a user that is part of the Administrator's group and restart this tool to create the database.

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

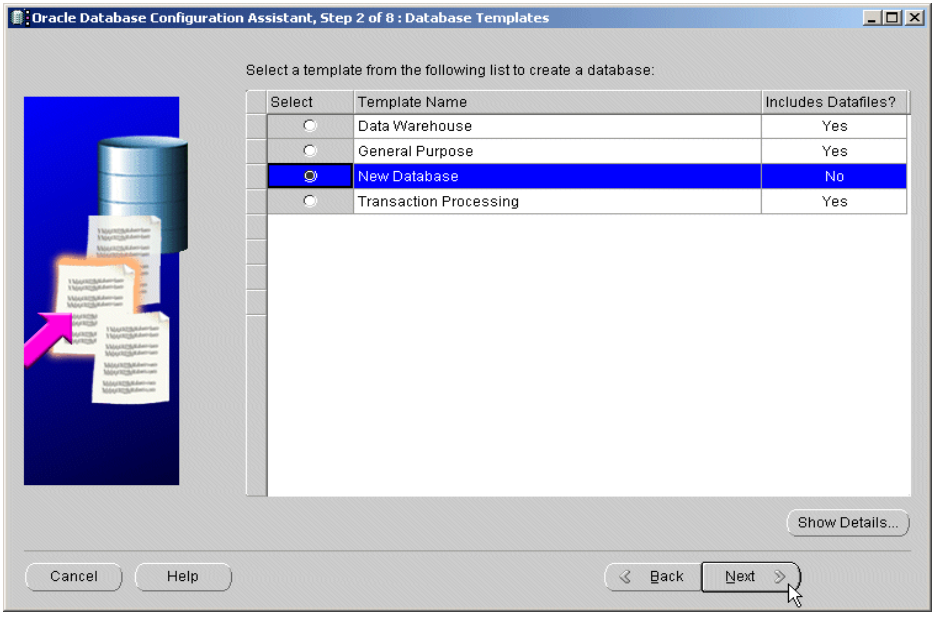
   The Oracle Database Configuration Assistant Welcome page appears.

2. Choose Next. A window appears prompting you to select an operation to perform.



3. Select Create a database, then Choose Next.

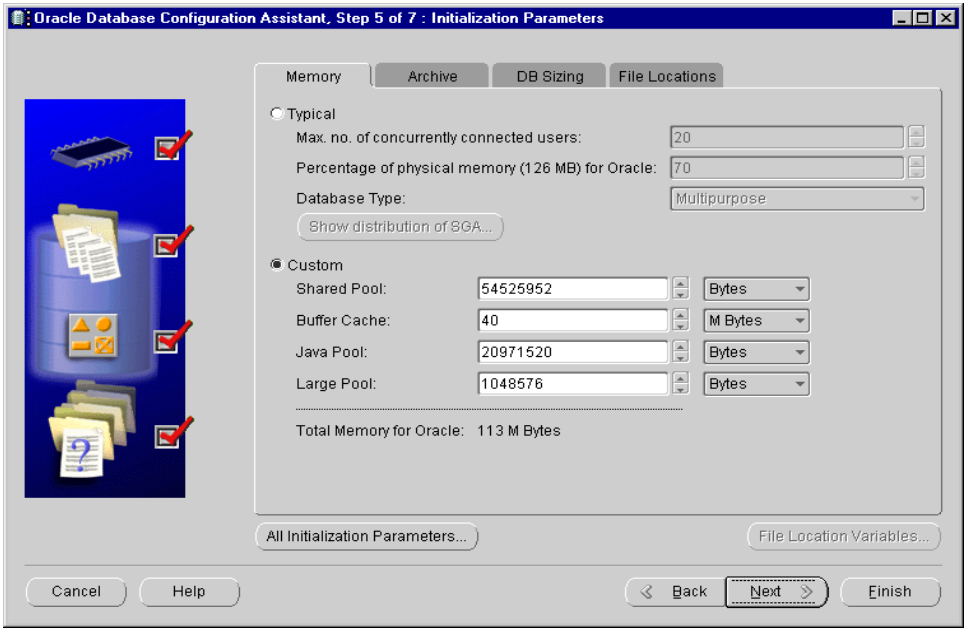
   The following window appears:



4. Select New Database, then Choose Next.
5. Enter a Global Database Name typically of the form `name.domain` and an Oracle **System Identifier (SID)**. Choose Next.
6. Select the features you want to configure for the new database. Choose from Oracle Spatial, Oracle OLAP Services Features, and several sample **schema**s. Choose Next.
7. Select a database connection option:
   - Dedicated Server Mode. Each Client connection to the database allocates a resource dedicated to that client only.
   - Shared Server Mode. Several Client connections share an allocated pool of resources.

   Choose Next.
8. Select a Typical or Custom Installation. A Typical installation lets you set:
   - Maximum number of connected users
   - Percentage of physical memory for Oracle
   - Database type

   A Custom installation lets you specify initialization parameter values and other options including:
   - Shared Pool size
   - Buffer Cache size
   - Java Pool size
   - Large Pool size
9. Choose Next.



10. Respond to instructions on each Oracle Database Configuration Assistant page, then Choose Next when you are ready to continue to the next page. When you get to the last page, Choose Finish to start the creation of the Oracle9*i* database.

Importing Sample Schemas

Sample Online Transaction Processing (OLTP) and data warehousing database schemas are available on the CD-ROM. After you finish creating your Oracle9*i* database with Oracle Database Configuration Assistant, you can import the appropriate sample schema.

If you selected multipurpose, you do not need to import the 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 appropriate schema:

```
SQL> CREATE USER sample_user IDENTIFIED BY password;

SQL> GRANT RESOURCE TO sample_user;

SQL> GRANT CONNECT TO sample_user;
```

   where *sample_user* is `sampleoltp` for the `OLTP` sample schema or `samplestar` for the data warehousing sample schema.

4. Exit SQL*Plus:

```
SQL> EXIT
```

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

6. Import the appropriate schema:

```
C:\oracle\ora90\assistants\dbca\samples> imp sample_user/password
FILE=sample.dmp FULL=y LOG=myimp.log
```

   where:

`C:\oracle`	is the *ORACLE_BASE* directory
`ora90`	is the *ORACLE_HOME* directory
sample_user	is `sampleoltp` for the OLTP sample schema or `samplestar` for the data warehousing sample schema
sample.dmp	is `source90.dmp` for the OLTP sample schema or `target90.dmp` for the data warehousing sample schema
`myimp.log`	import log file

See Also:

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

for more information on using Oracle Database Configuration Assistant

Delete a Database

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

Creating a Database Manually

This section describes how to create a new database manually. Oracle provides a sample database creation script, and a sample initialization parameter file with the database software files it distributes, both of which can be edited to suit your needs. If you have existing scripts you can use them to create a database manually or you can edit your existing script using the sample database creation script as a guide. There are a number of ways to create a database depending on if you want to:

- Make a copy of an existing database and remove the old database.
- Make a copy of an existing database and keep the old database.
- Create a new database when no database exists on your system.

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

Table 1–1 Database Creation Scenarios

Perform these tasks...	If you want to...		
	Copy an existing database to a new database, then remove the old database	**Copy an existing database to a new database, then keep the old database**	**Create a new database when no other database exists on the system**
Exporting an Existing Database	Yes	Only if you want to copy data from the existing database to the new database	Not applicable
Deleting Database Files	Yes	No	Not applicable
Modifying the init.ora file	Yes	Yes	Yes
Creating and Starting an Oracle Service	No	Yes	Yes
Putting the CREATE DATABASE Statement in a Script	Yes	Yes	Yes
Creating a Database	Yes	Yes	Yes

Table 1–1 Database Creation Scenarios

Perform these tasks...	If you want to...		
	Copy an existing database to a new database, then remove the old database	Copy an existing database to a new database, then keep the old database	Create a new database when no other database exists on the system
Importing a Database	Yes	Only if you want to import tables and other objects exported from the existing database	Not applicable
Updating the ORACLE_SID in the Registry	No	Only if you want to change the default `SID`	Yes
Backing Up the New Database	Yes	Yes	Yes

How to Create a Database

An example is used in the following sections to demonstrate how to create a database.

In this example, you will copy an existing database (the **starter database** with a `SID` of `orcl` located in the `C:\oracle\oradata\orcl` directory) to a new database with a database name and `SID` of `prod` located in the `C:\oracle\oradata\prod` directory.

You will delete the starter database `orcl` after you have created the `prod` database.

> **Note:** In this example, *`ORACLE_BASE`* is `C:\oracle`. See Oracle9i Database installation guide for Windows for more information on *`ORACLE_BASE`*.

Creating Directories

Create the following directories in which to put the administration and database files for the 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 only need to export an existing database if you want to copy its contents to a new database.

You can start the Export utility by using either parameter mode or interactive mode. However, parameter mode is the recommended 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 the Export utility prompt you for the information it needs.

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

Note: If you use parameter mode, the Export utility considers file names and directory names to be invalid if a blank space is present. Enclose the full path in the `file=` parameter in triple quotation marks. For example:

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

or

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

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

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 want to export. For example, if the database you want to export is the starter database `orcl`, enter the following at the MS-DOS command prompt. Note that there are no spaces around the equal sign (=) character.

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

2. Start the Export utility from the MS-DOS command prompt:

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

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

Deleting Database Files

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

To delete database files:

1. Shut down the starter database `orcl` at the MS-DOS command prompt:

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

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

   - `control01.ctl`
   - `control02.ctl`
   - `control03.ctl`
   - `indx01.dbf`
   - `Drsys01.dbf`
   - `CWMLITE01.DBF`
   - `EXAMPLE01.DBF`
   - `UNDOTBS01.DBF`
   - `system01.dbf`
   - `temp01.dbf`
   - `users01.dbf`
   - `redo01.log`
   - `redo02.log`
   - `redo03.log`
   - `tools01.dbf`

Modifying the init.ora file

If you are using the starter database `orcl` as the basis for your new database, copy the `init.ora` file:

`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, you cannot copy an initialization parameter file to use as the basis for your new `init.ora` file. However, you can use the sample initialization parameter file `initsmpl.ora` provided in the *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\admin\sample\pfile` directory as the basis for the `init.ora` file for the `prod` database.

If you use `initsmpl.ora` as the basis for the `init.ora` file, you must modify the following initialization parameters in the `init.ora` file, or you will not be able to start the `prod` database:

- `DB_NAME`
- `INSTANCE_NAME`
- `SERVICE_NAMES`
- `CONTROL_FILES`
- `BACKGROUND_DUMP_DEST`
- `USER_DUMP_DEST`

Modifying the `DB_FILES` initialization parameter is recommended to optimize performance. Table 1–2 describes the modification instructions.

Table 1–2 Modifying DB_FILES Initialization Parameters

Initialization Parameter	Modification Instructions
DB_NAME	This parameter indicates the name of the database and must match the name used in the CREATE DATABASE statement in "Putting the CREATE DATABASE Statement in a Script" on page 1-16. You give a unique database name to each database. You can use up to eight characters for a database name. The name does not need to match the SID of the database service. Set this parameter to: `DB_NAME=PROD.DOMAIN`
INSTANCE_NAME	Set this parameter to: `INSTANCE_NAME=prod.domain`
SERVICE_NAMES	Set this parameter to: `SERVICE_NAMES=prod.domain`
CONTROL_FILES	This parameter lists the control files of the database. You do not have the control files on your file system at this point, because the control files are created when you run the CREATE DATABASE statement. Ensure that you specify the complete path and filename, including drive letter. Set this parameter to: `CONTROL_FILES = ("C:\oracle\oradata\prod\control01.ctl", "C:\oracle\oradata\prod\control02.ctl", "C:\oracle\oradata\prod\control03.ctl")`
BACKGROUND_DUMP_DEST	Set this parameter to: `BACKGROUND_DUMP_DEST = C:\oracle\admin\prod\bdump`
USER_DUMP_DEST	Set this parameter to: `USER_DUMP_DEST = C:\oracle\admin\prod\udump`
DB_FILES	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. `DB_FILES=100`

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

Creating and Starting an Oracle Service

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

- Copy an existing database to a new database and keep the old database
- Create a new database when no other database exists on your system that you can copy

Before you create the database, first create a Windows NT 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 the ORADIM utility to create the service. After it has been created, the service starts automatically. See "Using the ORADIM Utility to Administer an Oracle Instance" on page 1-22 for information on how to use the ORADIM utility.

To create and start an Oracle service:

1. Run the ORADIM utility from the MS-DOS command prompt:

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

   Note that the previously created `init.ora` file is specified, with complete path, including drive name. You can check if the service is started in the services window of the Windows NT 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 want to 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 the clauses that you specified in the CREATE DATABASE statement or the initialization parameters that you have set.

> **Note:** Oracle Managed Files is a feature that can work with the CREATE DATABASE statement. Using Oracle-managed files simplifies the administration of an Oracle database. Oracle-managed files eliminate the need to directly manage the operating system files comprising an Oracle database. Using Oracle Managed Files can simplify the use of the CREATE DATABASE statement. 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.*

The following statement creates the database prod:

```
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;
```

Creating a Database

To use the SQL script to create a database:

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

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

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

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

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

2. Make PROD the current SID:

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

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

   ```
   C:\> SQLPLUS / NOLOG
   ```

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

   The password is the one that you used to create the service, with the `oradim -new` command in "Creating and Starting an Oracle Service" on page 1-16.

   The message *Connected* appears.

4. Turn on spooling to save the messages:

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

5. Run the `script_name.sql` script that you created in "Creating a Database" on page 1-16:

   ```
   SQL> @C:\oracle\ora90\rdbms\admin\script_name.sql;
   ```

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

Importing a Database

You can import the full export created in "Exporting an Existing Database" on page 1-12 into the new database.

You can also start the Import utility using parameter mode or interactive mode. Parameter mode is recommended, because interactive mode provides less 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 the command `imp SYSTEM/password` to begin an interactive session and let the Import utility prompt you for the information it needs.

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

Note: If you use parameter mode, the Import utility considers filenames and directory names to be invalid if there is a blank space. Enclose the full path in the FILE= parameter in triple quotation marks. For example:

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

or

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

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

To import a database:

- Run the 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, the Import utility tries to create those tablespaces with associated datafiles.

The easy solution is to ensure that both databases contain the same tablespaces. The datafiles do not have to be identical. Only the tablespace names are important.

Updating the ORACLE_SID in the Registry

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

1. Start the registry editor at the MS-DOS command prompt:

   ```
   C:\> regedt32
   ```

   The registry editor window appears.

2. Choose the `\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME0` subkey 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:** Oracle9i Database installation guide for Windows for more information on the subkey locations for **multiple Oracle homes**

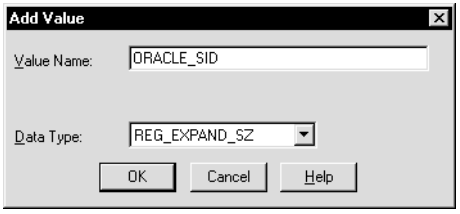
3. Locate the ORACLE_SID parameter on the right side of the registry editor window.
4. Double-click the parameter name and change the data to the new SID, which is prod in this example.

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

To create the ORACLE_SID parameter:

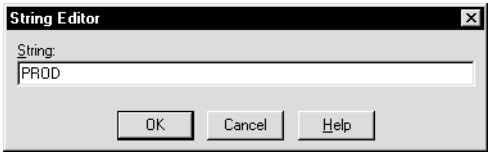
1. Choose Add Value from the Edit menu.

   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.

   The registry editor adds the ORACLE_SID parameter.

7. Choose Exit from the Registry menu.

   The registry editor exits.

Backing Up the New Database

> **Caution:** If anything goes wrong while operating the new database without a backup, 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 the 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 the `OracleServicePROD` service has stopped. If you do not do this, the backup may be useless as it was taken while data was being written to the datafiles.

2. Using the tool of your choice, back up the 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 to the database, such as switching the 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 the ORADIM Utility to Administer an Oracle Instance

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

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

If you have installed an Oracle database service on Windows 2000, 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 the `ORADIM.LOG` file in the *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\database` directory.

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

Also, 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:

1. Remove the line `sqlnet.authentication_services=(NTS)` from `SQLNET.ORA` or set `sqlnet.authentication_services=(NONE)` in `SQLNET.ORA`.
2. Start the database manually after the Oracle database service has started using SQL*Plus connecting as `SYSDBA`.
3. Start the service as a specific user:
    1. Choose Start > Settings > Control Panel > Services. The Services dialog box appears.
    2. Select the service you want to start.
    3. Click the Startup button. The Service dialog box appears.
    4. Click This account and specify the username and corresponding password.

Note: If you are using Windows 2000, in the Control Panel, double-click Administrative Tools first, then double-click Services.

The following sections describe the ORADIM utility commands and parameters. Note that each command is preceded by a dash (-). Table 1–3 shows how to get a list of ORADIM utility parameters.

Table 1–3 ORADIM Utility Parameters

To get a list of ORADIM utility parameters and descriptions...	
Use this syntax:	`oradim -? | -H | -HELP` **Note:** Specifying `oradim` without any options also returns a list of `oradim` parameters and descriptions.
Example:	`C:\> oradim -?`

Creating an Instance

Table 1–4 describes how to use the ORADIM utility to create an instance.

Table 1–4 Using the ORADIM Utility to Create an instance

To create an instance...		
Use this syntax:	`oradim -NEW -SID` *`SID`* `| -SRVC` *`service_name`* `-SHUTTYPE srvc | inst | srvc, inst` `[-MAXUSERS` *`number`*`][-STARTMODE auto | manual][-PFILE "`*`filename`*`"]`	
Example to create an instance called PROD:	`C:\> oradim -NEW -SID prod -INTPWD mypassword1 -STARTMODE auto -PFILE "C:\oracle\admin\prod\pfile\init.ora"`	
Syntax description:	■ `-NEW`	Indicates that you want to create a new instance. This is a mandatory parameter.
	■ `-SID` *`SID`*	The name of the instance you want to create. You must specify either this parameter or the -SRVC parameter described below.
	■ `-SRVC` *`service_name`*	The name of the service you want to create (`OracleService`*`SID`*). You must specify either this parameter or the -SID parameter described above.
	■ `-MAXUSERS` *`number`*	The number of users defined in the password file. The default is 5.

Table 1–4 Using the ORADIM Utility to Create an instance

■ `-STARTMODE auto, manual`	Indicates whether to start the instance automatically or manually at startup. The default setting is `manual`.
■ `-PFILE filename`	The `init.ora` file to be used with this instance. Ensure that you specify the complete path name of this file, including drive letter.
■ `-SHUTTYPE srvc, inst`	Indicates whether to stop the service or the instance. Both can be specified. This is a mandatory parameter.

Starting an Instance

Table 1–5 describes how to use the ORADIM utility to start an instance.

Table 1–5 Using the ORADIM Utility to Start an Instance

To start an instance...		
Use this syntax:	`ORADIM -STARTUP -SID` *`SID`* `[-USRPWD` *`user_pwd`*`][-STARTTYPE SRVC | INST | SRVC, INST] [-PFILE "FILENAME"]`	
Example to start an instance called `puma`**:**	`C:\> ORADIM -STARTUP -SID puma -STARTTYPE SRVC -PFILE "C:\ORACLE\ADMIN\PROD\PFILE\init.ora"`	
Syntax description:	■ `-STARTUP`	Indicates that you want to start an instance that already exists. This is a mandatory parameter.
	■ `-SID` *`SID`*	The name of the instance you want to start. This is a mandatory parameter.
	■ `-USERPWD` *`user_pwd`*	The password.
	■ `-STARTTYPE` *`SRVC, INST`*	Indicates whether to start the service or the instance. One or both values can be specified. If not specified, the registry is checked for the current setting.

Stopping an Instance

Table 1–6 describes how to use the ORADIM utility to stop an instance.

Table 1–6 Using the ORADIM utility to Stop an Instance

To stop an instance:		
Use this syntax:	`ORADIM -SHUTDOWN -SID` *`SID`* `[-USRPWD` *`user_pwd`*`] [-SHUTTYPE SRVC | INST | SRVC, INST] [-SHUTMODE A | I | N]`	
Example to stop an instance called `puma`**:**	`C:\> ORADIM -SHUTDOWN -SID puma -SHUTTYPE SRVC INST`	
Syntax description:	■ `-SHUTDOWN`	Indicates that you want to stop an instance. This is a mandatory parameter.
	■ `-SID` *`SID`*	The name of the instance you want to stop. This is a mandatory parameter.

Table 1–6 Using the ORADIM utility to Stop an Instance

■ `-USERPWD` *`user_pwd`*	The password.
■ `-SHUTTYPE` *`srvc, inst`*	Indicates whether to stop the service or the instance. One or both values can be specified. If not specified, 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, normal is the default mode.

Modifying an Instance

Table 1–7 describes how to use the ORADIM utility to modify an instance.

Table 1–7 Using the ORADIM utility to Modify an Instance

To modify an instance...		
Use this syntax:	`ORADIM -EDIT -SID` *`sid`* `[-NEWSID` *`NEWSID`*`] [-STARTMODE AUTO | MANUAL][-PFILE "`*`FILENAME`*`"]`	
Example to modify an instance called PROD:	`C:\> ORADIM -EDIT -SID PROD -NEWSID LYNX -INTPWD MYCAT123 -STARTMODE AUTO -PFILE "C:\ORACLE\ADMIN\LYNX\PFILE\INIT.ORA"`	
Note:	You can modify an existing instance, in this example `PROD`, to change such values as the instance name, the password, the startup mode, and the number of users.	
Syntax description:	■ `-EDIT`	Indicates that you want to modify an instance. This is a mandatory parameter.
	■ `-SID` *`SID`*	The name of the instance you want to modify. This is a mandatory parameter.
	■ `-NEWSID` *`NEWSID`*	The new instance name. This is an optional parameter.
	■ `-STARTMODE AUTO, MANUAL`	Indicates whether to start the instance automatically or manually at startup. The default setting is `MANUAL`.
	■ `-PFILE FILENAME`	The `INIT.ORA` file to be used with this instance. Ensure that you specify the complete path name of this file, including drive letter.

Deleting an Instance

Table 1–8 describes how to use the ORADIM utility to delete an instance.

Table 1–8 Using the ORADIM utility to Delete an Instance or Service

To delete an instance or service...		
Use this syntax:	`ORADIM -DELETE -SID SIDA` `ORADIM -DELETE -SVRC SVRCA`	
Example to delete an instance called PROD:	`C:\> ORADIM -DELETE -SID PROD`	
Syntax description:	■ `-DELETE`	Indicates that you want to delete an instance or service.
	■ `SIDA`	The name of the `SID` you want to delete.
	■ `SVRCA`	The name of the service you want to delete

2

Postinstallation Configuration Tasks

This chapter describes some of the 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 Spatial
- Shared Server Support
- Advanced Replication

Note: The 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, your directory paths will differ. See Multiple Oracle Homes and Optimal Flexible Architecture in the *Oracle9i Database Getting Started for Windows* for more information.

Oracle *inter*Media

This section describes the components of Oracle *inter*Media.

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.

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.

Oracle *inter*Media Audio, Video, Image, and Locator Configuration Responsibilities

Review Table 2–1 to determine how to configure Oracle *inter*Media Audio, Video, Image, and Locator.

Table 2–1 How to Configure Oracle Intermedia Products

If You...	Then...
Installed the Oracle9*i* Enterprise Edition Typical installation	No manual configuration is required. All tasks described in "Configuring Oracle interMedia Audio, Video, Image, and Locator" on page 2-3 are automatically performed.
Installed the Oracle9*i* Enterprise Edition	Oracle Database Configuration Assistant starts at the end of installation. If you select the following option: ▪ Custom installation, then select the Create new database option
Manually copy your Oracle8*i* `listener.ora` and `tnsnames.ora` files into your Oracle9*i* network directory	You must modify the `tnsnames.ora` and `listener.ora` network configuration files on your server to enable **external routine** calls to work and *inter*Media to function properly. Follow the tasks in *Oracle9i Net Services Administrator's Guide.*
Want to use the demos	Perform the tasks described for the configuration of the Oracle *inter*Media sample demos below.
Want to build the Oracle *inter*Media Image sample demonstration	Install a C compiler.

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

To configure Oracle *inter*Media Audio, Video, Image, and Locator:

1. Start SQL*Plus:

   ```
   C:\> SQLPLUS / NOLOG
   ```

2. Connect to the database with the SYS account:

   ```
   SQL> CONNECT SYS/password AS SYSDBA
   ```

3. Start the database (if necessary):

   ```
   SQL> STARTUP
   ```

4. Run the `ordinst.sql` script:

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

5. Run the `iminst.sql` script:

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

6. Exit SQL*Plus:

```
SQL> EXIT
```

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

To configure Oracle *inter*Media Audio demos:

1. Go to the *ORACLE_BASE**ORACLE_HOME*\ord\aud\demo directory.
2. Follow the instructions in the `readme.txt` file.

To configure Oracle *inter*Media Video demonstrations:

1. Go to the *ORACLE_BASE**ORACLE_HOME*\ord\vid\demo directory.
2. Follow the instructions in the `readme.txt` file.

To configure Oracle *inter*Media Image demonstrations:

1. Go to the *ORACLE_BASE**ORACLE_HOME*\ord\img\admin directory.
2. Read the `readme` file in this directory for instructions on configuring the demo.
3. Make the *inter*Media Image demos for a Microsoft C compiler by entering:

```
C:\> make
```

An additional demonstration resides in the following location:

*ORACLE_BASE**ORACLE_HOME*\ord\img\demo\vc\imgsamp\simpimg
In order to build and run the demonstration, you must first modify the MAKEFILE to adapt it to your environment.

To configure Oracle *inter*Media Locator demonstrations:

1. Go to the *ORACLE_BASE**ORACLE_HOME*\md\demo\geocoder directory.
2. The `nh_cs.sql` file contains the sample data that can be loaded into Oracle9*i*. `geohttp.sql` and `geolocat.sql` are examples that show the use of Locator functionality. `geoindex.sql` contains examples of data indexes created by using the Locator.

Oracle Text Configuration Responsibilities

Review Table 2–2 to determine your Oracle Text postinstallation tasks.

Table 2–2 Oracle Text postinstallation tasks

If You...	Then...
Installed Oracle Text from the CD-ROM, and you do not have a previous release of Oracle Text (formerly called *inter*Media Text) installed	See the description below this table.
Installed Oracle Text from the CD-ROM, and you do have a previous release of Oracle Text (formerly called *inter*Media Text) installed	See *Oracle Text Application Developer's Guide.*
Migrated your database, you may need to configure Oracle Net for external routines. Otherwise, Oracle Text may not work. In any case other than migration, Oracle Net should be configured correctly by default to work with Oracle Text.	See *Oracle Text Application Developer's Guide* and *Oracle Text Reference.*
Are indexing formatted documents such as Microsoft Word	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.*

If one of the following is true, the Oracle9*i* database is already configured for use with Oracle Text:

- The database is a starter database that you created by installing Oracle9*i* Enterprise Edition, Oracle9*i* Standard Edition or Oracle9*i* Personal Edition.
- The database is a starter database that you created by performing the following sequence of steps:

  1. Installed Oracle9*i* Enterprise Edition, Oracle9*i* Standard Edition, or Oracle9*i* Personal Edition.
  2. Selected Oracle9*i* Server in the *Available Product Components* window.
  3. Selected the General Purpose creation type.

  See Also: See Chapter 1, "Postinstallation Database Creation" and the Oracle9i Database installation guide for Windows for more information about creating a starter database.

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

If none of these are true, you must configure the Oracle database for use with Oracle Text by doing one of the following:

- Using Oracle Database Configuration Assistant
- Configuring Manually

Using Oracle Database Configuration Assistant

You can use Oracle Database Configuration Assistant to configure an Oracle9*i* database for use with Oracle Text at the time you create the database or later. Table 2–3 describes how to configure an Oracle9*i* database

Table 2–3 Configure an Oracle9i database for Use with Oracle Text

To configure...	Do this...
At a later time	**1.** Select Configure database options in a database. **2.** Select the database that you want to modify when prompted. **3.** 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 the Oracle Text **data dictionary** tables and then creating the `ctxsys` username and the Oracle Text data dictionary tables themselves.

To create a tablespace for the **Oracle Text** data dictionary tables:

1. Start SQL*Plus:

   ```
   C:\> sqlplus / nolog
   ```

2. Connect as SYS:

   ```
   Enter user-name: SYS/password AS SYSDBA
   ```

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

   ```
   SQL> CREATE TABLESPACE tablespace_name DATAFILE 'ORACLE_BASE\oradata\
   db_name\dr01.dbf' SIZE 80m;
   ```

To create the `ctxsys` username and the Oracle Text data dictionary tables:

1. Connect AS SYSDBA:

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

2. Run the `dr0csys.sql` script to create the `ctxsys` username:

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

where:

- `password` is the password that you want to use for the `ctxsys` username.
- `default_tablespace_name` is the default tablespace for the Oracle Text data dictionary tables. Set the default tablespace to the value of `tablespace_name` in step 3 of the instructions "To create a tablespace for the Oracle Text data dictionary tables:".
- `temporary_tablespace_name` is the temporary tablespace for the Oracle Text data dictionary tables. Set the temporary tablespace to the value of `tablespace_name` in step 3 of the instructions "To create a tablespace for the Oracle Text data dictionary tables:".

3. Connect as `ctxsys`:

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

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

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

5. 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;
```

6. Exit SQL*Plus:

```
SQL> EXIT
```

Oracle Spatial

Oracle Spatial makes the 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. The locations of cities, roads, and political boundaries are projected onto a two-dimensional display or piece of paper, preserving the relative positions and relative distances of the objects.

Review Table 2–4 to determine your configuration responsibilities:

Table 2–4 Configuration Responsibilities

If You...	Then...
Installed Oracle Spatial through the Oracle9*i* Enterprise Edition	No manual configuration is required. All Oracle Spatial configuration tasks described under "Configuring Oracle Spatial" are automatically performed.
Installed both Oracle Spatial and Oracle9*i* Server together through the Oracle9*i* Enterprise Edition or Oracle9*i* Standard Edition Custom installation	Oracle Database Configuration Assistant starts at the end of installation. If you select the following: ■ Custom installation, and then select Create new database Oracle Database Configuration Assistant prompts you about whether or not you want Oracle Spatial to be automatically configured.
Installed Oracle Spatial during a separate installation from Oracle9*i* Enterprise Edition	You must manually configure Oracle Spatial by either: ■ Performing the Oracle Spatial configuration tasks described under "Configuring Oracle Spatial" ■ Starting Oracle Database Configuration Assistant and selecting Configure database options in a database.

Configuring Oracle Spatial

1. Start SQL*Plus at the MS-DOS command prompt:

   ```
   C:\> sqlplus / nolog
   ```

2. Connect to the database with the SYSDBA account:

   ```
   Enter user-name: SYS/password AS SYSDBA
   ```

3. Start the database (if necessary):

   ```
   SQL> STARTUP
   ```

4. Run the `ordinst.sql` script:

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

5. Connect to the database as the SYSTEM user:

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

6. Run the `mdinst.sql` script:

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

7. Exit SQL*Plus:

```
SQL> EXIT
```

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

Shared Server Support

Oracle 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.

Table 2–5 describes the differences between dedicated server mode and shared server mode.

Table 2–5 Dedicated Server Mode and Shared Server Mode Differences

Mode	Description
Dedicated server mode	The Oracle9*i* database allocates a resource dedicated to serving only that one client connection. This mode is best used in the following environments: ■ Warehousing environment. ■ A small number of users will connect to your Oracle9*i* database.

Table 2–5 Dedicated Server Mode and Shared Server Mode Differences

Mode	Description
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. This mode is best used in the following environments: ■ Online transaction processing (OLTP) environment ■ A large number of users will simultaneously connect to your database ■ You want to use Oracle Net features such as connection pooling, connection multiplexing, and load balancing ■ Managing and using system resources to a high degree is important ■ Predictable and fast database connection times are very important. This may be very important for Web applications. **Note:** This mode is required for Oracle9*i* JVM to function properly.

See Also: *Oracle9i Net Services Administrator's Guide* for more information about the shared server mode

There are two types of shared server support:

- Internet Inter-ORB Protocol (IIOP) clients
- Two-task Oracle Net clients

These types are independent of each other. In other words, you can have any of the following combinations:

- Both types of support are enabled
- One type of support is enabled, and the other type of support is disabled

Table 2–6 describes how the current configuration of your Oracle9*i* database depends on how the database was installed.

Table 2–6 Configuration of the Database Based on Installation Method

If installed through...	Then the configuration is...
Oracle9*i* Enterprise Edition installation	Shared server mode for IIOP clients and dedicated server mode for two-task Oracle Net clients
Typical option of Oracle Database Configuration Assistant	If you selected Oracle9*i* JVM, the mode is shared server mode for IIOP clients. Dedicated server mode for two-task Oracle Net clients, unless you perform the following sequence of steps: 1. Run Oracle Database Configuration Assistant. 2. Select Create a database. 3. Select New database. 4. Select Shared Server Mode. Choose the Edit Shared Connections Parameters button. 5. On the Basic tab of the Shared Server Mode page enter 20 or more for the maximum number of server processes. This creates a database in shared server mode for two-task Oracle Net clients.
Custom option of Oracle Database Configuration Assistant	Dedicated server mode or shared server mode, depending on what you select when prompted by Oracle Database Configuration Assistant.

Enabling Shared Server Support for IIOP Clients

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

Enabling Shared Server Support for Two-Task Oracle Net Clients

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

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

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

   The Oracle Database Configuration Assistant Welcome page appears. Choose Next.

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

   A dialog box prompts you to select the initialization parameter file to use.
9. Select the appropriate file and choose OK.

   Your initialization parameter file is modified.
10. 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 Oracle 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. Choose 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 the **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 want to add Advanced Replication to an Oracle9*i* database that was not previously configured with this feature.

Configuring Advanced Replication consists of the following steps:

- Task 1: Checking Tablespace and Rollback Segment Requirements
- Task 2: Checking Initialization Parameters
- Task 3: 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*

Task 1: Checking Tablespace and Rollback Segment Requirements

The following are recommended tablespace and rollback segment requirements for Advanced Replication:

Tablespace/Rollback Segment	Requirement
SYSTEM	At least 20 MB of free space is required for replication packages. Replication triggers and procedures are stored here.
UNDO TABLESPACE	At least 10 MB of free space.
RBS	At least 5 MB of free space.
TEMP	At least 10 MB of free space.
USERS	No specific requirement.

Task 2: Checking Initialization Parameters

Certain initialization parameter values must be set or added to the `init.ora` file (recommended values are included) if you use Advanced Replication.

Master Site

Add these initialization parameters to the master site.

Parameter Name	Recommended Value
JAVA_POOL_SIZE	50 MB
DISTRIBUTED_LOCK_TIMEOUT	300 seconds
DISTRIBUTED_TRANSACTIONS	5
GLOBAL_NAMES	TRUE
OPEN_LINKS	4
PROCESSES	Add 9 to current value
JOB_QUEUE_PROCESSES	2 (depends on number of n-way sites)

Materialized View Sites

Add this initialization parameter for materialized view sites.

Parameter Name	Recommended Value
JOB_QUEUE_PROCESSES	2

Task 3: Monitoring Data Dictionary Tables

If you use Advanced Replication and intend to set up a large number of replicated objects, 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 the storage parameters to accommodate the 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

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

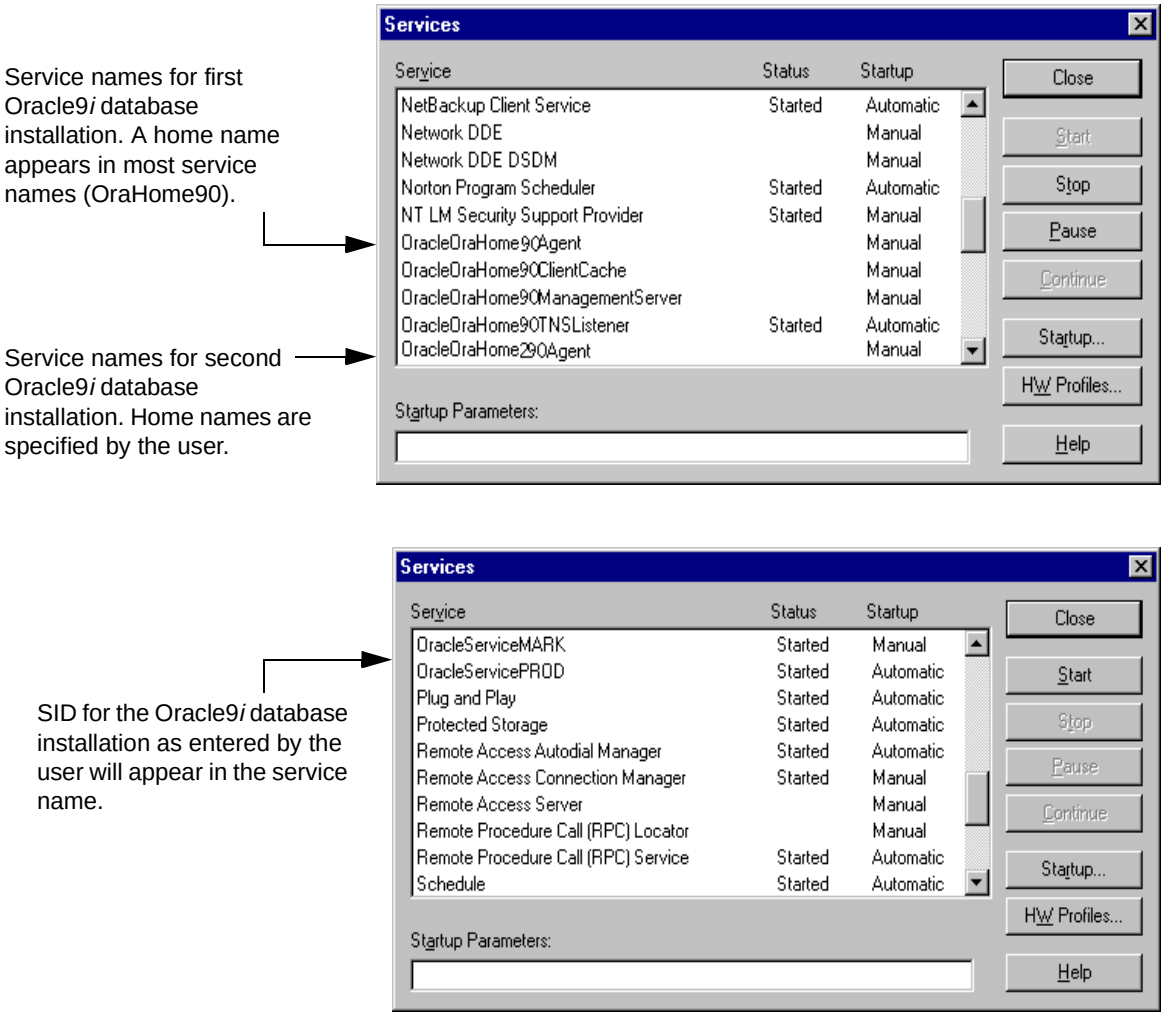
Oracle9*i* for Windows lets you have multiple, active Oracle home directories on a single computer.

The Database Tools Overview chapter of *Oracle9i Database Getting Started for Windows* describes this feature. Multiple Oracle homes affect the naming conventions for Oracle services. As you perform installations into Oracle home directories:

- You must accept the default Oracle home name provided or specify a different name for each Oracle home directory during installation, which is added to most service names
- 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:

Figure 3–1 Services Dialog Boxes



Available Oracle Services

Depending on the products that you have installed, a number of Oracle services are started when you restart your Windows NT computer. A user with a non-system account must have local administrative rights to run services on a Windows NT computer. The two main Oracle services are described in Table 3–1.

Table 3–1 Oracle Services

Service Name	Description
`OracleService`*`SID`*	Created for the database instance *`SID`*. An Oracle instance is a logical term that refers to: ■ An Oracle service called `OracleService`*`SID`* ■ A database Each Oracle instance must have a (`SID`). A `SID` is a unique name for an Oracle database instance that can be up to 64 alphanumeric characters in length. For example, if the `SID` for the Oracle9*i* database is `orcl`, it is appended to the service `OracleService`. The instance name is the same as the value of the `ORACLE_SID` registry configuration parameter.
`Oracle`*`home_name`*`TNSListener`	Listens for and accepts incoming connection requests from client applications. Automatically starts when the Windows NT computer restarts. The *`home_name`* that displays in this service name is the value you entered in the Name field on the File Locations dialog box of Oracle Universal Installer.

Table 3–2 lists additional Oracle services related to specific products or database features that are also available:

Table 3–2 Additional Oracle Services

Service Name	Description
`Oracle`*`home_name`*`Agent`	Listens for and responds to job and event requests sent from the Oracle Enterprise Manager console.

Services for network products that are available are described in Table 3–3:

Table 3–3 Oracle Services for Network Products

Service Name	Description
`Oracle`*`home_name`*`TNSListener` or, for example: `Oracle`*`home_name`*`TNSlistener`*`lsnr`*	The service name, if you are using the default network listener name `listener`. *`lsnr`* is the nondefault network listener name. It is only created if the following command has been run: `lsnrctl START` *`lsnr`*
`Oracle`*`home_name`*`ClientCache`	Used for the Client Cache Service.
`Oracle`*`home_name`*`CMAdmin` `Oracle`*`home_name`*`CMan`	Used for the product Oracle Connection Manager.

See Also: *Oracle9i Net Services Administrator's Guide* for general information on Oracle network services

Starting Oracle Services

Oracle services must be started for you to use the Oracle9*i* database and its products.

Start Oracle services in either of three ways:

- From the Control Panel
- From the MS-DOS command prompt
- From the Oracle Administration Assistant for Windows NT

Note: You can start the 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.

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 you want to start in the list, select it, and choose Start.

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

4. Choose Close to exit the Services dialog box.

> **Note:** If you are using Windows 2000, in the Control Panel, double-click Administrative Tools first, then double-click Services.

To start Oracle services from the MS-DOS command prompt:

Enter the following command to start an Oracle service at the MS-DOS command prompt:

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

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

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

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

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

3. Choose Start Service

   This starts the service `OracleServiceORCL`.

Stopping Oracle Services

On occasion (for example, if you want to re-install the Oracle9*i* database), you must stop Oracle services. Stop Oracle services in either of three ways:

- From the Control Panel
- From the MS-DOS command prompt
- From the Oracle Administration Assistant for Windows

> **Note:** You can stop the 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.

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. Choose OK.

   `OracleService`*`SID`* is stopped.

To stop Oracle services from the MS-DOS command prompt:

Enter the following command to stop an Oracle service at the MS-DOS command prompt:

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

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

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

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Oracle Administration Assistant for Windows NT.

2. Right-click the *`sid`*.

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

3. Choose Stop Service

   This stops the service `OracleServiceORCL`.

Auto-starting Oracle Services

You can start Oracle services whenever the Windows computer is restarted. Use the Service dialog box to configure when and how the Oracle9*i* database is started.

To automatically start the Oracle9*i* database each time you restart Windows:

1. Choose Start > Settings > Control Panel.

   The Control Panel window appears.

2. Double-click Services.

   The Services dialog box appears.

3. Select the service `OracleServiceSID` and choose the Startup button.

   The Service dialog box appears.

4. Choose Automatic from the Startup Type field.

5. Choose OK.

6. Choose Close to exit the Services dialog box.

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

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Oracle Administration Assistant for Windows NT.

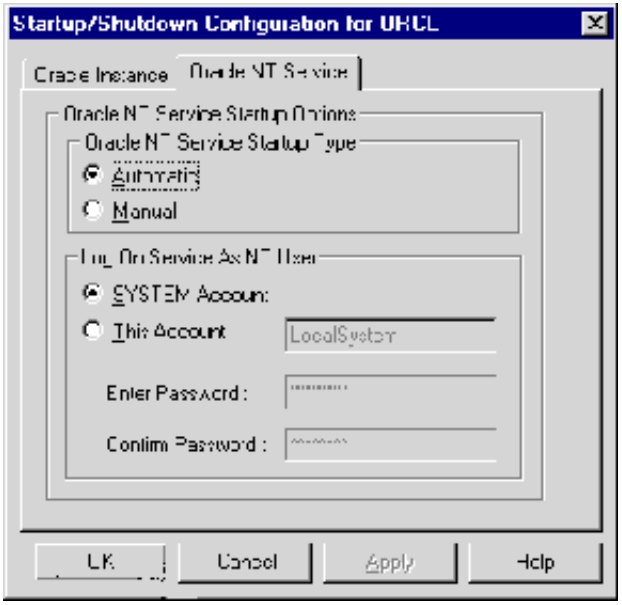
2. Right-click the *`sid`*.

   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. Choose Apply.
7. Choose OK.

Starting and Shutting Down a Database with SQL*Plus

These instructions assume that a database instance has been created.

> **Note:** The 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, 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 MS-DOS command prompt:

```
C:\> SQLPLUS / NOLOG
```

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

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

4. Follow the instructions below:

If You Want to...	Then Enter...
Start a database with the default parameter file	`SQL> STARTUP` This command uses the default `init.ora` file located in the *`ORACLE_BASE`*`\ADMIN\`*`db_name`*`\pfile` directory.
Start a database with a file other than the default parameter file	`SQL> STARTUP PFILE=`*`path\filename`* This command uses the `init.ora` file specified in *`path\filename`*. This example starts the database using a file named `init2.ora` in `C:\ora90\admin\orcl\pfile`: `SQL> STARTUP PFILE=C:\ora90\admin\orcl\pfile\init2.ora`
Stop the database	where *`mode`* is one of the following: ■ `NORMAL` The database waits for all currently-connected users to disconnect and disallows any new connections before shutting down. This is the default mode. ■ `IMMEDIATE` The database terminates and rolls back active transactions, disconnects clients, and shuts down. ■ `ABORT` The 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 the 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 the Oracle9*i* database by starting or stopping the service OracleServiceSID in the Control Panel. This automated procedure is equivalent to using the STARTUP or SHUTDOWN commands or manually entering the following:

If You...	These Commands are Performed...
Start OracleService*SID*	C:\> oradim -STARTUP -SID *SID* -USERPWD *password* -STARTTYPE SRV,INST
Stop OracleService*SID*	C:\> oradim -SHUTDOWN -SID *SID* -USERPWD *password* -SHUTTYPE srv,inst -SHUTMODE -i

To start the database by starting OracleService*SID*:

1. Review or set the following registry parameters. Note that the ORADIM utility, when used to create or edit instances, automatically sets these values in the registry.

Parameter	When Set to true, this Parameter...
ORA_*SID*_AUTOSTART	The default value, starts the database when OracleService*SID* is started.
ORA_*SID*_PFILE	Sets the full path to the init.ora parameter file. The default path is *ORACLE_BASE*\ADMIN*db_name*\pfile\init.ora.

See *Oracle9i Database Getting Started for Windows* for instructions on adding and editing registry parameters.

The exact location in which to set these parameters is determined by the number of Oracle home directories on your computer:

If You Have...	Then Add These Parameters in...
One home directory	HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME0
Additional directories	HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*ID* where *ID* is incremented for each additional Oracle home directory on your computer.

2. Choose Start > Settings > Control Panel.

The Control Panel window appears.

3. Double-click Services.

   The Services dialog box appears.

4. Select OracleService*SID* and choose Start.

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

To shut down the database by stopping OracleService*SID*:

1. Set either of the following parameters to true in the registry:

Parameter	When Set to true, this Parameter...
ORA_SHUTDOWN	Enables the selected Oracle9*i* database to be shut down. This includes any database in the current Oracle home.
ORA_*SID*_SHUTDOWN	Shuts down the Oracle9*i* database identified by the *SID* value.

   If either is set to false (the default setting), it is possible to shut down the database by stopping OracleService*SID*, but this is not recommended because it results in an abnormal shutdown of the database.

   The exact location in which to set these parameters is determined by the number of Oracle home directories on your computer.

If You Have...	These Parameters are Located in...
One home directory	HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME0
Two or more home directories	HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME*ID* where *ID* is incremented for each additional Oracle home directory on your computer.

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

2. Set the following optional parameters to appropriate values in the registry:

Parameter	Description
ORA_*SID*_SHUTDOWNTYPE	Set to a (abort), i (immediate), or n (normal) to indicate the database shutdown mode. The default mode is i (immediate) if you do not set this parameter.
ORA_*SID*_SHUTDOWN_TIMEOUT	Sets the maximum time to wait before the service for a particular SID stops.

3. Choose Start > Settings > Control Panel.

   The Control Panel window appears.

4. Double-click Services.

   The Services dialog box appears.

5. Select OracleService*SID* and choose Stop.

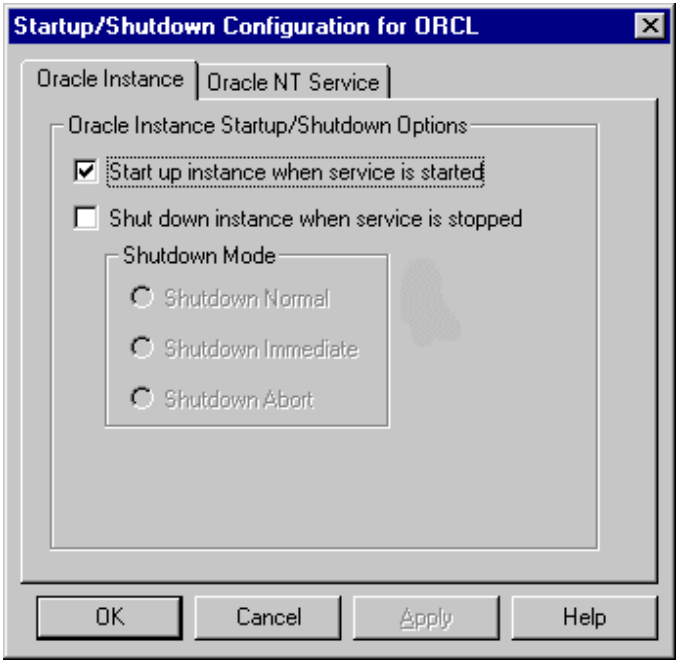
   This automatically starts the ORADIM utility, which issues the -SHUTDOWN command in the mode indicated by ORA_*SID*_SHUTDOWNTYPE, and shuts down your Oracle9*i* database.

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

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Oracle 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 or select Shut down instance when service is stopped.



Running Multiple Instances

To run multiple instances, ensure that you have already created each instance and started the services for each instance using the ORADIM utility or the Services dialog box in Windows NT. You then run multiple instances by starting each of the instances using SQL*Plus.

To run multiple instances:

1. Ensure that you have already created each instance.
2. Ensure that you started the services for each instance using the ORADIM utility or the Services dialog box of the Windows NT Control Panel.
3. Set the `ORACLE_SID` configuration parameter at the MS-DOS command prompt to the SID for each instance you want to run:

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

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

   ```
   C:\> SQLPLUS / NOLOG
   ```

5. Connect AS SYSDBA:

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

6. Start up the database with the new instance:

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

   where *ORACLE_BASE* is `c:\oracle` by default (unless you changed it during installation) and *db_name* is the name of the instance.

Creating Password Files

Use the Password Utility to create password files. The Password Utility is automatically installed with the Oracle9*i* Utilities. Password files are located in the *ORACLE_BASE**ORACLE_HOME*\DATABASE directory 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. The following example describes how to perform a local connection.

To create a password file:

1. Create a password file with the Password utility:

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

   The essential elements of a password file are:

Element	Description
SID	Identifies the database instance.
FILE	Specifies the password filename.
PASSWORD	Sets the password for the SYS account.
ENTRIES	Sets the maximum number of entries in the password file. This corresponds to the maximum number of distinct users allowed to connect to the database with the SYSDBA and **SYSOPER** DBA privileges simultaneously.

2. Set the `init.ora` file parameter REMOTE_LOGIN_PASSWORDFILE to `exclusive` or `shared`. Definitions for all possible values are described below:

Element	Description
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 the ORA_*SID*_PWFILE parameter. If a value is unspecified, it looks in the registry for the value of the ORA_PWFILE parameter, which points to a file containing the usernames, passwords, and privileges. If that is not set, it uses the default of *ORACLE_BASE**ORACLE_HOME*\DATABASE\PWD*sid*.ORA.
shared	Specifies that multiple instances can use the password file (for example, a Real Application Clusters environment). 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 releases. The Oracle9*i* database looks for the same files as it does when the value is exclusive. This is the default value.
none	Specifies that the Oracle9*i* database ignores the password file and that privileged users are authenticated by the Windows operating system. none is the default setting.

3. Start SQL*Plus:

```
C:\> SQLPLUS / NOLOG
```

4. Connect AS SYSDBA:

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

5. Start the Oracle9*i* database:

```
SQL> STARTUP
```

6. Grant appropriate privileges to each user who must perform database administration. For example:

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

   If successful, 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.

7. Connect to the Oracle9*i* database with administrative privileges for scott:

```
SQL> CONNECT scott/tiger AS SYSDBA
```

You are connected to the Oracle9*i* database.

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

Viewing Password Files

The password file is automatically hidden. This section describes two ways of viewing the password file:

- From the MS-DOS command prompt:
- From Windows NT Explorer:

Table 3–4 lists commands for viewing the password file from the MS-DOS command prompt:

Table 3–4 Viewing the Password File from the MS-DOS Command Prompt

To...	Enter...

See the password file	`C:\oracle\ora90\database> attrib`
	The password file is displayed as `PWD`*`sid`*`.ora`:
	`A H C:\oracle\ora90\database\PWD`*`sid`*`.ora`

Make the password file visible	`C:\oracle\ora90\database> attrib -H PWD`*`sid`*`.ora` **Note:** The password file must be visible before you can move or copy it.
Hide the password file again	`C:\oracle\ora90\database> attrib +H PWD`*`sid`*`.ora`

To see the password file from Windows NT Explorer:

1. Open Windows NT Explorer.
2. Open the folder to view.
3. Choose Folder Options from the View main menu.
4. Choose the View tab.

5. Follow the instructions below:

To...	Choose...
See the password file	Show all files.
Hide the password file	Do not show hidden files.

Deleting Password Files

This section describes how to delete a password file.

To delete a password file:

1. Make the password file visible at the MS-DOS command prompt or in Windows NT Explorer by following the instructions in section "Viewing Password Files" on page 3-17.
2. Delete the password file based on whether you made it visible at the MS-DOS command prompt or in Windows NT Explorer.

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 the 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 `init.ora` file on the server computer.
2. Set `DBLINK_ENCRYPT_LOGIN` equal to `true`.
3. Set the `ORA_ENCRYPT_LOGIN` configuration variable 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, the Oracle9*i* database encrypts the password before sending it to the remote database. If the connection fails, the failure is noted in the audit log. The Oracle9*i*

database then checks if either of these parameters is set to `false`. If so, the Oracle9*i* database attempts the connection again using an unencrypted version of the password. If the connection is successful, the success is noted in the audit log, and the connection proceeds.

Note: Releases prior to release 7.1 do not support encrypted passwords. If you are connecting to an earlier version of the Oracle database, you must set the initialization parameter `DBLINK_ENCRYPT_LOGIN` to `false` for the connection to succeed.

Creating Control, Data, and Log Files on Remote Computers

Although it is possible for Oracle to access database files on remote computers using Universal Naming Convention (UNC), it is not recommended or supported because of performance and network reliability concerns.

UNC is a PC format for specifying the location of resources on a local area network. UNC uses the following format:

```
\\server-name\shared-resource-pathname
```

For example, to access the file `system01.dbf` in the directory `C:\oracle\ora90\oradata\orcl` on the shared server `argon`, you reference the file as:

```
\\argon\oracle\ora90\oradata\orcl\system01.dbf
```

Note that the location of archive log files cannot be specified using UNC. If you set the `LOG_ARCHIVE_DEST_`*n* initialization parameter to a UNC specification, the 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
```

Ensure that you set the `LOG_ARCHIVE_DEST_`*n* initialization parameter to a mapped drive.

Note: 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 files and datafiles did not.

Archiving Redo Log Files

Your Oracle9*i* database is created in NOARCHIVELOG mode if you installed your database through the Typical installation. If you created your database through the Custom option of Oracle Database Configuration Assistant, 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 the database from both instance and disk failure.

This section describes how to change the archive mode to ARCHIVELOG and enable automatic archiving. See "Managing Archived Redo Logs" of *Oracle9i Database Administrator's Guide* for complete descriptions of the ARCHIVELOG and NOARCHIVELOG modes.

Task 1: Change the Archive Mode to ARCHIVELOG

To change the archive mode to ARCHIVELOG:

1. Start SQL*Plus at the MS-DOS command prompt:

   ```
   C:\> SQLPLUS / NOLOG
   ```

2. Connect to the Oracle9*i* database with the SYSDBA username:

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

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

   ```
   SQL> SHUTDOWN
   ```

4. 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 the 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;
```

9. Continue to "Task 2: Enable Automatic Archiving".

Task 2: Enable Automatic Archiving

To enable automatic archiving:

1. Open the *ORACLE_BASE*\ADMIN*db_name*\pfile\init.ora file.
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:** The default size of the Redo logs is approximately 100 MG.

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

Format	Description	Example
%%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.	*SID*0001.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.	*SID*0001.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.	*SID*1.ARC
%%ORACLE_SID%%%sARC	Specifies the log sequence number. The number is not padded. The default value is one with no range limit on characters.	*SID*1.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 the ORADEBUG Utility

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

To start the ORADEBUG utility:

1. Start SQL*Plus from the MS-DOS command prompt and connect to the database as SYSDBA. For example:

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

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

```
SQL> ORADEBUG
```

The 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
```

The output from most debug commands is written to a **trace file**. Trace files are created in the directory specified by the `init.ora` 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. If you want to find the location of your trace file, enter the following at the SQL*Plus prompt:

```
SQL> ORADEBUG TRACEFILE_NAME
```

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

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

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

Table 4–1 Database Monitoring Tools

This Tool...	Enables You To...
Oracle for Windows NT Performance Monitor	Monitor database objects, such as CPU usage, buffer cache, and background processes.
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 Oracle threads.

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 the Windows NT Performance Monitor, except it has been preloaded with Oracle9*i* database performance elements.

On each computer, you can view the behavior of objects, such as the 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.

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

The 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 the 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", of the *Oracle9i Database Getting Started for Windows* guide, for instructions on how to use the 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 the `Hostname` registry value so it points to another computer specified in the `tnsnames.ora` file.

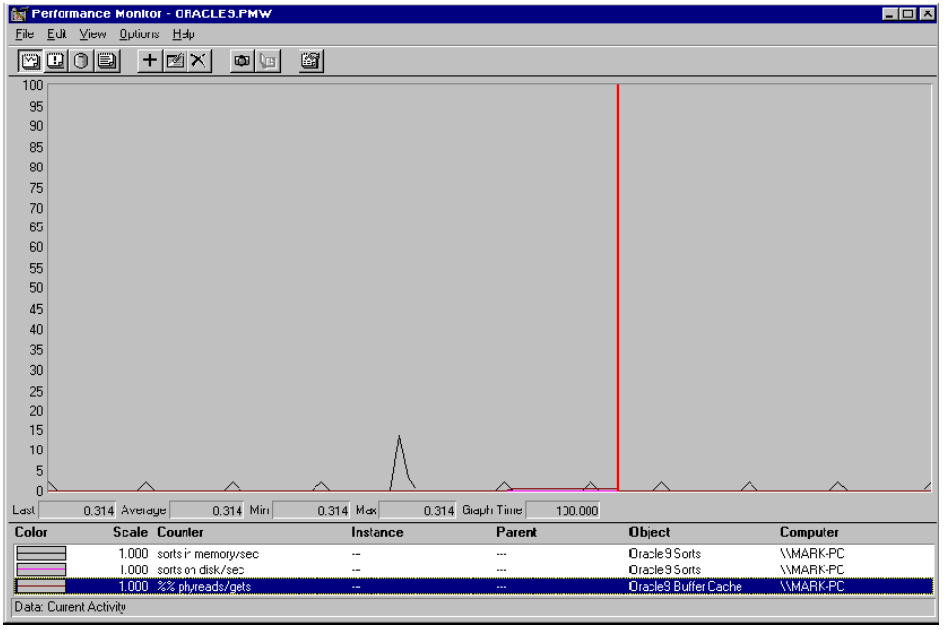
Accessing Oracle for Windows NT Performance Monitor

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 the Performance Monitor window displaying the Chart View:

Figure 4–1 Oracle Performance Monitor Window



The Oracle for Windows NT Performance Monitor has four views you can choose from the View menu. Table 4–2 describes these views:

Table 4–2 Oracle Performance Monitor Views

View	Description
Chart View	Displays database activity in real-time.
Alert View	Lets you know when certain minimum performance criteria are not being met, or maximum criteria are being exceeded.

Table 4–2 Oracle Performance Monitor Views

View	Description
Log View	Maintains continuous records on performance.
Report View	Saves information about specific criteria.

Monitoring Oracle9*i* Objects

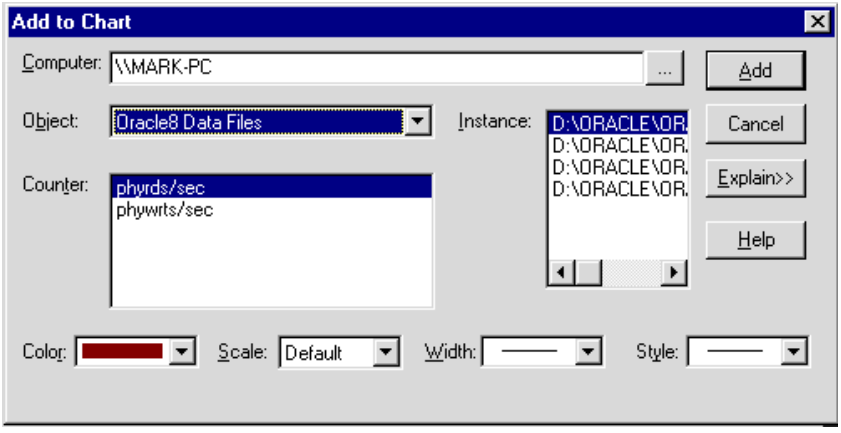
For each view (Chart, Alert, Log, and Report), you can decide on the objects you want to monitor and save those settings to a file. When an object is chosen, it is assigned a counter, a color, and added to the status bar at the bottom of Oracle for Windows NT Performance Monitor.

To add objects to a view:

1. Choose Add To (Chart, Alert, Log, Report) from the Edit menu.

   The Add to (Chart, Alert, Log, Report) dialog box appears.

   Below is the Add to Chart dialog box. Note the corresponding dialog boxes for the other views are different.



2. Select the objects you want to monitor, then choose Add.

   Below are the elements of the Add to Chart dialog box. The other views' dialog boxes have similar features.

Element	Description
Computer list box	Specify the computer you want to monitor.
Object drop-down list box	Select an object to monitor. **Note:** If no data or Oracle9*i* objects appear, either the database is not running, or an invalid host string or password has been entered. If the database is not started, exit Oracle for Windows NT Performance Monitor, start the database, and restart Oracle for Windows NT Performance Monitor.
Counter list box	Select a counter (or multiple counters) for the object you have selected. Note that the contents of the Counter box change depending upon your selection in the Object box. If you want details on how a counter works, highlight the counter and choose Explain.
Instance box	Select an instance for this counter.
Color box	Choose a color for the display of the selected counter.
Scale box	Choose the scale at which you want to display the counter.
Width box	Specify the width of the line on the graph.
Style box	Choose a different style for your graph line.

3. Choose 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*. These measures are defined in *ORACLE_BASE*`\`*ORACLE_HOME*`\dbs\perf.ora`. Table 4–3 shows the Oracle9*i* objects and their associated counters. For additional information on these objects, see *Oracle9i Database Performance Guide and Reference*.

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

Table 4–3 Oracle9i Objects and Counters

Object	Counter	Description
Oracle9*i* Buffer Cache	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	redo log space requests	The value of this counter must be near zero. If this value increments consistently, 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	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 above this threshold while your application is running, increase the amount of memory available to the data dictionary cache. To increase the memory available to the cache, increase the value of the initialization parameter `SHARED_POOL_SIZE`. (See *Oracle9i Database Performance Guide and Reference* for more detailed information on tuning memory allocation in the Oracle9*i* database.) This value is not time-derived.
Oracle9*i* Library Cache	reloads / pins %	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	■ phyrds / sec ■ phywrts / sec	Disk contention occurs when multiple processes try to access the same disk simultaneously. There are many ways of reducing disk contention, depending on the results from monitoring disk activity. Some corrective actions include: ■ Distributing I / O ■ Separating datafiles and redo log files ■ Separating tables and indexes ■ Striping table data These values are time-derived.
Oracle9*i* DBWR stats1		These counters are helpful in tuning the Buffer Cache.
	■ buffers scanned / sec	Buffers scanned / sec is the number of buffers the DBWR scanned per second. The buffers scanned are on the LRU (Least Recently Used) list.
	■ LRU scans / sec	LRU scans / sec is the number of times the DBWR scanned the (Least Recently Used) buffer list per second.

Table 4–3 Oracle9i Objects and Counters

Object	Counter	Description
Oracle9*i* DBWR stats2		These counters are helpful in determining how much work the DBWR has been requested to perform.
	■ timeouts / sec	Timeouts / sec is the number of times the DBWR timed-out per second. The DBWR is on a three second timeout interval. If the DBWR has not been posted within a three second interval, it times out.
	■ checkpoints / sec	Checkpoints / sec is the number of checkpoint messages processed by the database writer per second. Whenever a checkpoint occurs, the DBWR must be messaged (posted) to "write dirty buffers to disk."
Oracle9*i* Dynamic Space Management	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, it may be necessary to determine the cause. Examine the recursive calls statistic through the dynamic performance table V$SYSSTAT.
Oracle9*i* Free List	free list waits / requests %	Contention for free lists is reflected by contention for free data blocks in the buffer cache. You can determine if contention for free lists is reducing performance by querying V$WAITSTAT. If the number of free list waits for free blocks is greater than 1% of the total number of requests, consider adding more free lists to reduce contention.
Oracle9*i* Sorts	■ sorts in memory / sec ■ sorts on disk / sec	The default sort area size is adequate to hold all the 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 want to increase the 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 dialog box, either:

- The database is not running
- An invalid host string or password has been entered in the registry

The `operf.log` file located in *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\dbs` contains error messages about Oracle for Windows NT Performance Monitor.

To resolve this problem:

1. Check the *ORACLE_BASE**ORACLE_HOME*\dbs\operf.log file for error messages.
2. Resolve the problem as follows:
   - If the log file indicates an invalid host string or password, check the registry for correct values for Hostname, Password, and Username. See HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet of the *Oracle9i Database Getting Started for Windows* for further information about these values.
   - If the database is not started, exit Oracle for Windows NT Performance Monitor and restart the database.
3. Restart Oracle for Windows NT Performance Monitor.

Using the Event Viewer

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

Accessing the Event Viewer

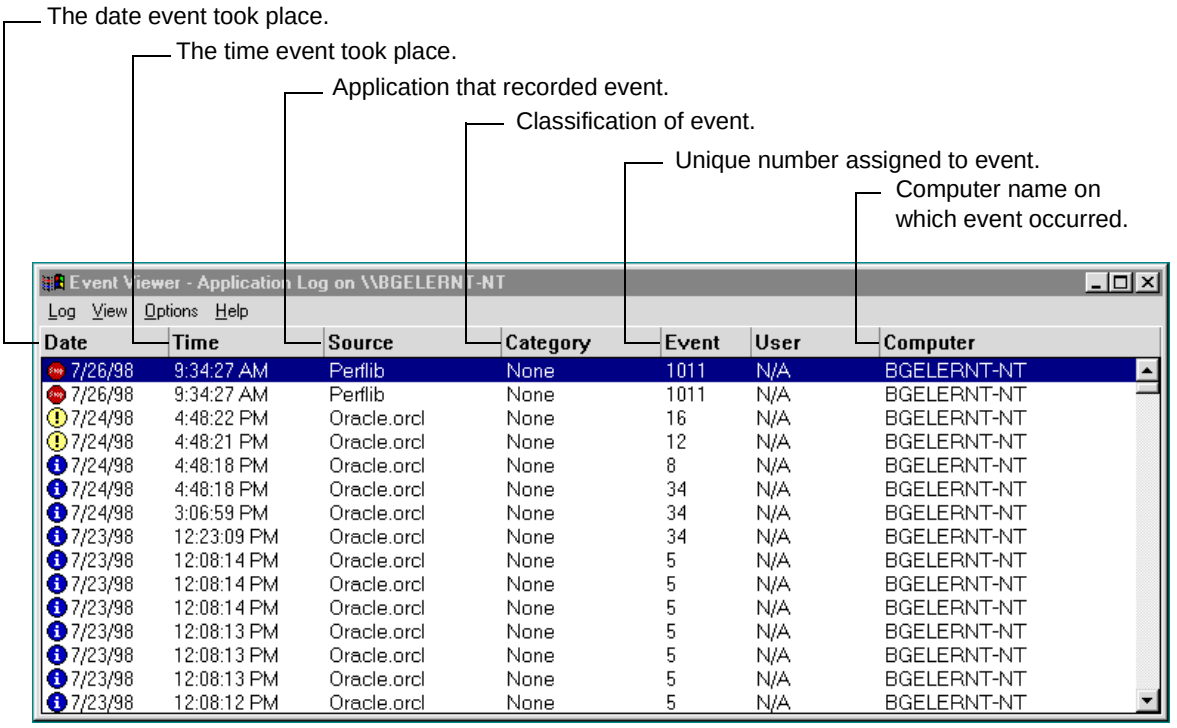
To access the Event Viewer:

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.

Figure 4–2 Application View window



Reading the Event Viewer

Table 4–4 describes the icons beside each event and the type of event.

Table 4–4 Event Viewer Icons

Icon	Event Type	Description
Red (stop sign)	Error	Indicates an error. Always check these icons.
Blue (informational)	Information	Indicates a noncritical system event. You can ignore these icons unless you want to track a specific event.
Yellow (exclamation point)	Warning	Indicates a special event, such as the termination of an instance or the shutdown of services. Investigate these icons, but they are usually noncritical.

Oracle9*i* for Windows NT events display with a source of Oracle.orcl. Oracle.orcl consists of the following event IDs described in Table 4–5:

Table 4–5 Oracle.orcl Event IDs

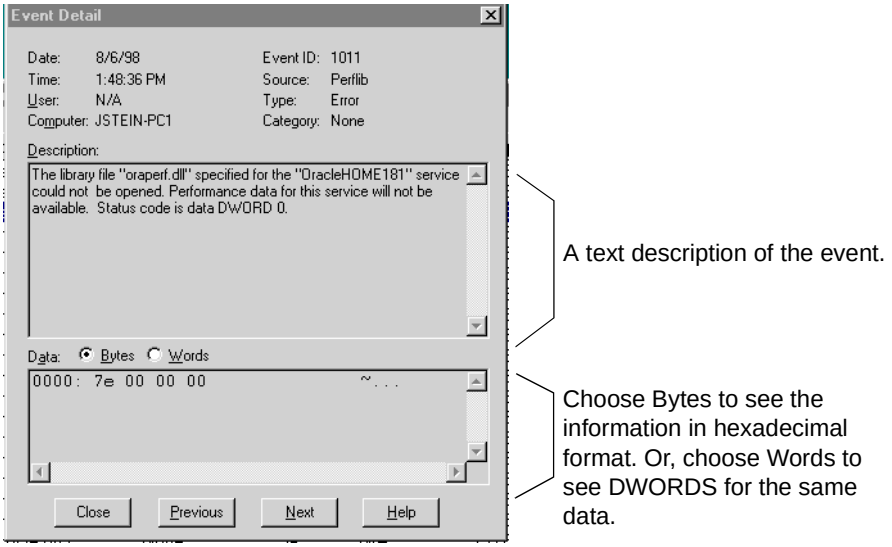
Event ID	Description
IDs other than 34	Specifies general database activities, such as an instance being started or stopped.
34	Specifies an audit trail event. These events are recorded if the `AUDIT_TRAIL` parameter is set to `db` (`true`) or `os` in the `init.ora` file. The OS option enables system wide auditing and causes audited records to be written to the Event Viewer. The `db` option enables system wide auditing and causes audited records to be written to the database audit trail (the `SYS.AUD$` table). Some records, however, are written to the Event Viewer.

Using the Event Viewer

To use the Event Viewer:

1. Look at the icons.
2. Double-click an icon to analyze (especially red icons).

The Event Detail dialog box appears with more information about the selected event:



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

Managing the Event Viewer

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

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. Choose OK.

   You are returned to the Event Viewer.

Caution: Audit information cannot be spooled to a file. The AUDIT_FILE_DEST parameter is not supported in Windows NT and should not be added to the init.ora file.

Using Trace and Alert 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 the background_dump_dest parameter is set in the init.ora initialization parameter file. If BACKGROUND_DUMP_DEST is set, the trace files are stored in the directory specified. If the parameter is not set, the trace files are stored in the *ORACLE_BASE**ADMIN**db_name*\bdump directory.

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:

- *sid*DBWR.TRC
- *sid*SMON.TRC

where *sid* represents the name of the instance.

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

The alert file contains important information about error messages and exceptions that occur during database operations. Each Oracle9*i* for Windows NT 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.

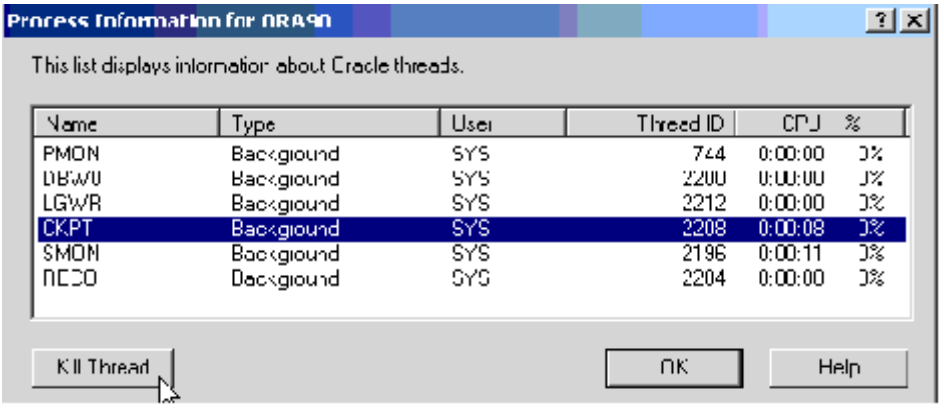
For example, when automatic archiving of redo logs is halted because no disk space is available, a message is placed in the alert file. The alert file is the first place to check if something goes wrong with the database and the cause is not immediately obvious.

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

Viewing Threads Using the Oracle Administration Assistant for Windows NT

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

1. Choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Oracle Administration Assistant for Windows.
2. Right-click the *sid*, where *sid* is a specific instance name, such as `orcl`.
3. Choose Process Information.
4. View information on appropriate threads. If you want to terminate a thread, select the thread you want to terminate.
5. 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:

Overview

Compared to UNIX, Windows NT Server offers considerably fewer settings that can be adjusted to tune the operating system. This reduces the ability of systems administrators to optimize Windows NT Server performance, but helps to make Windows NT Server easier to use than some operating systems.

There are still ways, however, to make Windows NT Server a better application server environment for the Oracle9*i* database. Most of these operating system specific procedures have the effect of reserving more system resources for the Oracle9*i* database, such as CPU, memory, and disk I/O. These procedures are described in this chapter. In addition, the Oracle9*i* database is a high-performance database management system that effectively uses the resources within your Windows NT computer. In general, the Windows NT computer that is running your Oracle9*i* database 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 both 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 there exist significant resources to accommodate all this activity.

Note: The information described in this chapter is specific to Oracle9*i* database installations on Windows NT Server, and not on Windows NT Workstation. This information is also applicable to Oracle9*i* database installations on Windows NT Server 4.0, Enterprise Edition. Windows NT Server 4.0, Enterprise Edition includes the capability for using additional products, such as the Microsoft Cluster Server, which is required if you want to implement Oracle Fail Safe (OFS).

Reduce Priority of Foreground Applications on the Server Console

Interactive foreground applications running on Windows NT Server are given priority over background processes. This is the setting provided by default during the installation of Windows NT Server. In order to prevent foreground applications on the server console from taking excessive processor time away from the Oracle9*i* database, remove the priority for foreground applications.

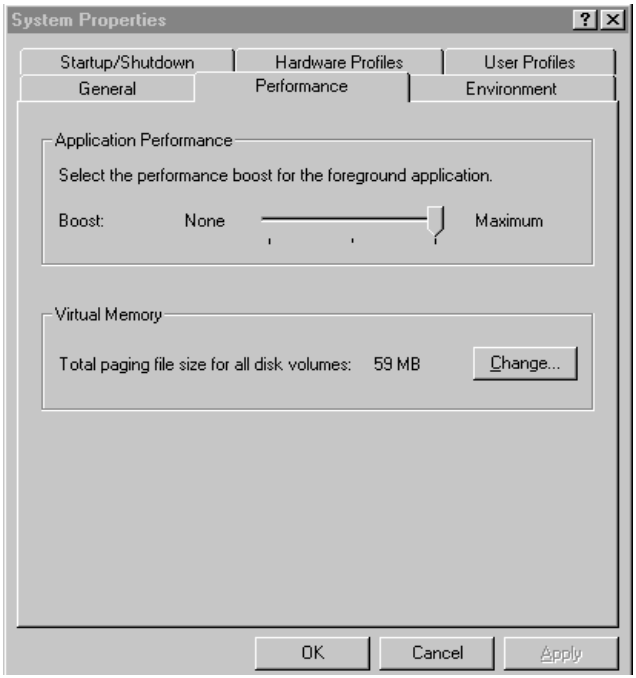
To reduce the priority of foreground applications:

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. Choose OK.
5. Exit the Control Panel.

Configure Windows NT Server to Be an Application Server

Windows NT Server is a 32-bit operating system capable of addressing a 4 GB memory space (that is, 2^{32} = 4,294,967,296 bytes = 4 GB). Half of this addressable memory space is reserved for system services and the file cache. The other half is addressable by user programs, such as the Oracle9*i* database.

Very few Windows NT Servers are equipped with 4 GB of RAM, so virtual memory plays an important role in Windows NT Server. The 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 the Oracle9*i* database) or if a large number of applications are running concurrently, the combined memory requirements of the applications may exceed the capacity of physical memory.

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

Table 5–1 Windows NT Server Memory Pools

Category	Pools	Approximate % of Total Memory
System Area	Kernel and other system services	9%
	File Cache	41%
User Area	Paged Memory	50%

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

Windows NT Server is set by default to perform as a file and print server with a large file cache. Reset the server memory model for network applications so that the file cache is reduced and more physical memory is available for the Oracle9*i* database.

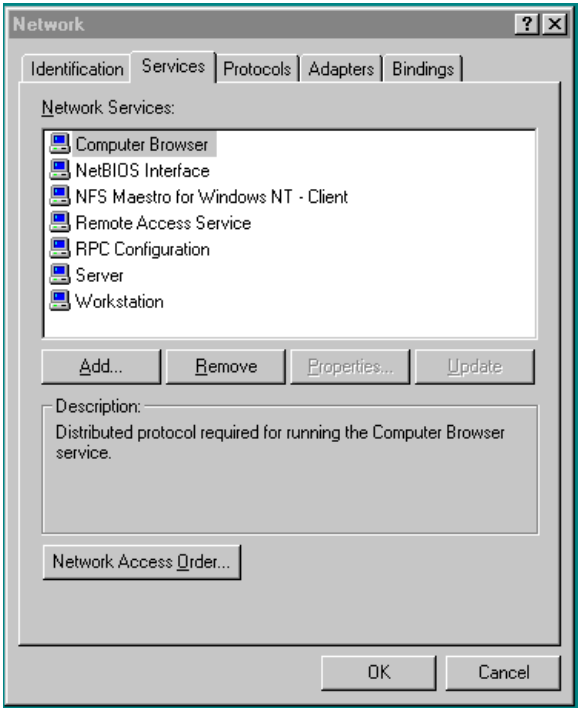
To configure Windows NT Server to be an applications server:

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

   The Network dialog box appears.

2. Choose the Services tab.

   The following information appears:



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

Disable Unnecessary Services

Once the file cache has been significantly reduced in size (as described in section "Configure Windows NT Server to Be an Application Server" on page 5-4), retrieve additional physical memory for the Oracle9*i* database by disabling services not needed for core operating system functionality.

To disable unnecessary services:

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

   The Services dialog box appears.

2. Scroll through the list of services and identify any unnecessary services. This is a partial list of services that can or cannot be disabled:

You Can Disable These Services	Do Not Disable These Services
License Logging Service	Alerter
Plug and Play	Computer Browser
Remote Access Autodial Manager	EventLog
Remote Access Connection Manager	Messenger
Remote Access Server	OracleService*sid*
Telephony Service	Oracle*home_name*TNSListener
	OracleStart*sid* (for releases 8.0.6 and earlier)
	Remote Procedure Call (RPC) Service
	Server
	Spooler
	TCP/IP NetBIOS Helper
	Workstation

Note: Consult with your systems administrator to find out if there are additional services that can be disabled.

3. Select the service.
4. Choose Startup.

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

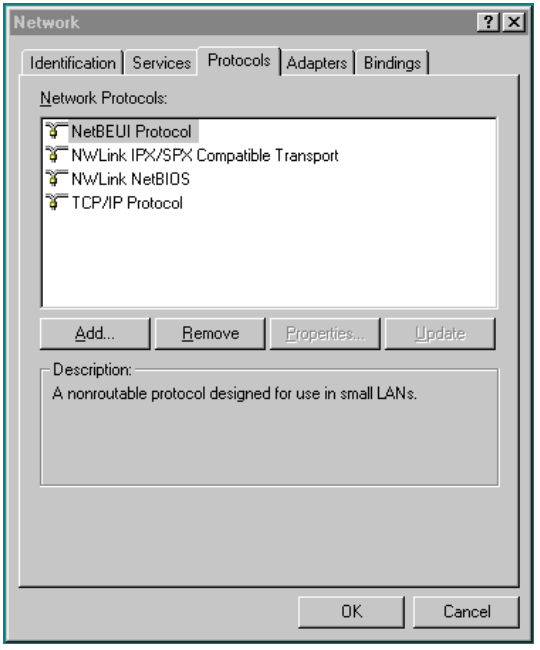
Remove Unused Network Protocols

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

To remove unnecessary network protocols:

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

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



3. If multiple protocols are installed, choose one, preferably TCP/IP, for use as the sole network protocol, unless others are necessary for server functionality.
4. Select unneeded network protocols and choose Remove.
5. Choose Yes in the confirmation dialog box until only the necessary protocols remain.
6. Exit the Control Panel.

Reset the Network Protocol Bind Order

If multiple protocols must be installed on the server, then prioritize the bindings so that the protocol most frequently used by the Oracle9*i* database is given the highest priority.

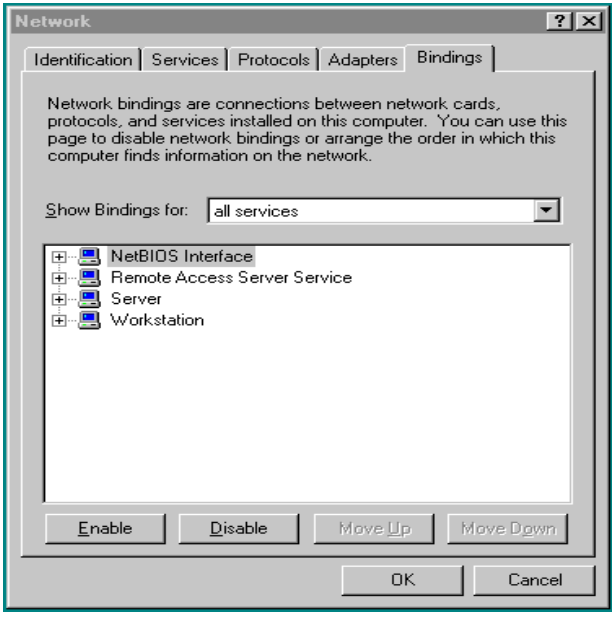
To reset the network protocol bind order:

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

   The Network dialog box appears.

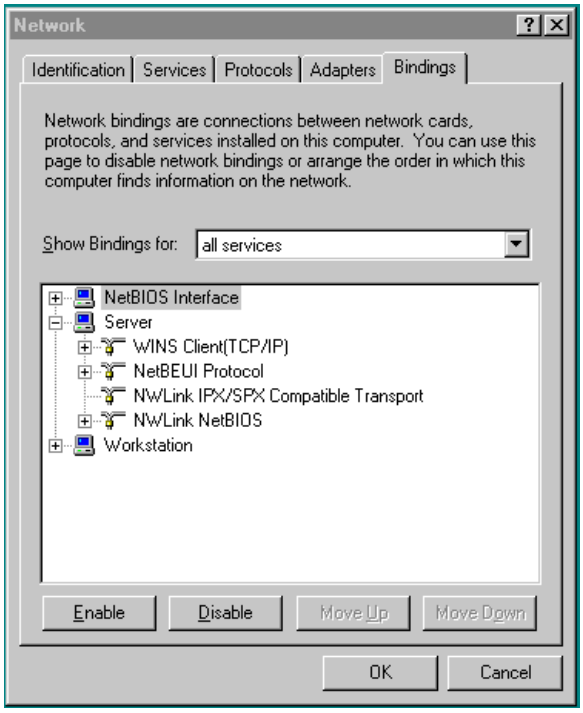
2. Choose the Bindings tab.

   The Bindings tab displays:



3. Select all services from the Show Bindings drop-down 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 the protocol.
6. Choose 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 the Oracle9*i* database to the top of the list.
8. Choose the OK button to save the changes.
9. Exit the Control Panel.
10. Reboot 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 the 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 (as a self-extracting archive) may be downloaded from:

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

To install a Service Pack:

1. Download the Service Pack version you want.
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 from those 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 the copying of files from the 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 the `UPDATE.EXE` (or the self-extracting archive) program.
2. Follow the appropriate prompts.

Use Hardware or Operating System Striping

Now that hard disk drives are relatively inexpensive, Windows NT Server should use logical volumes comprised of striped physical disks.

Compared to solid state CPU and memory speeds, mechanical hard disk drives are extremely slow. 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, 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. Each are described in Table 5–2, along with a fourth RAID level, RAID 0+1:

Table 5–2 RAID levels in Oracle9i Database Installations

RAID Level	Description	Read Penalty	Write Penalty
0 (Disk striping)	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 the "number of physical drives in the array" chunks, 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 is that there is no fault tolerance and if one disk in the logical volume fails, the whole volume fails and must be restored from a backup.	1:1 (1 I/O per read request)	1:1 (1 I/O per write request)
1 (Disk mirroring)	Enables fault tolerant disk mirroring (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 established.	1:1 (may benefit from split reads on some controllers) (for example, lower seek times when the controller knows which mirror contains the fastest accessed data)	2:1 (writing to two sources)

RAID Level	Description	Read Penalty	Write Penalty
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.	1:1 (may benefit from split reads on some controllers) (for example, lower seek times when the controller knows which mirror contains the fastest accessed data)	2:1
5 (Distributed Data Guarding (Disk striping with parity)	Enables distributed data guarding, eliminating the costly need 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 the data, although access times are greatly reduced due to the on-the-fly rebuilding of bytes from the 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 the parity information.	1:1	4:1 (2 reads / 2 writes during parity calculation)

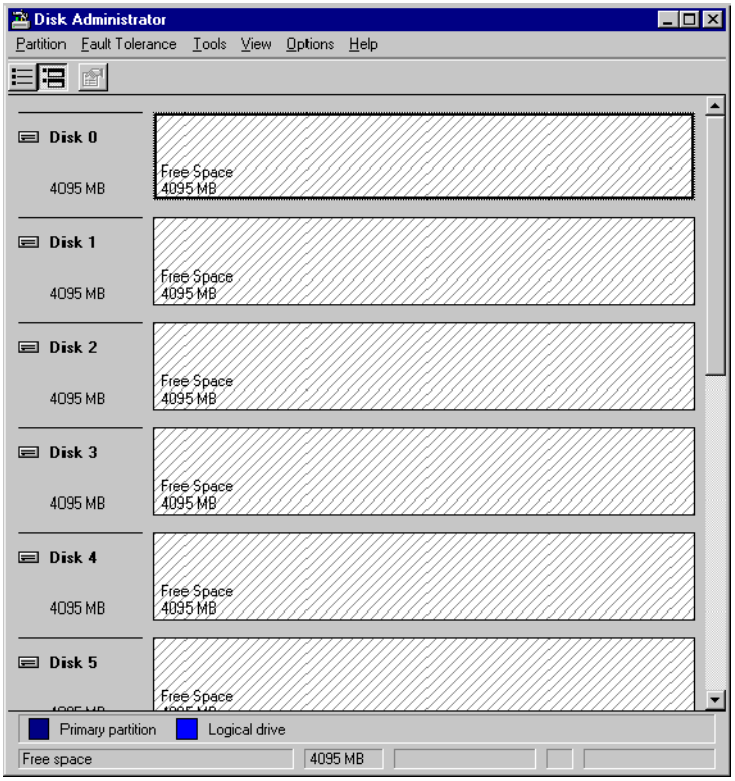
Data striping is achieved at the hardware level through the use of a RAID controller or, less efficiently, at the operating system level through Windows NT Server stripe sets. For a performance Windows NT Server configuration, data striping without parity (RAID-0 or equivalent) may be the best choice.

This example demonstrates the 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:

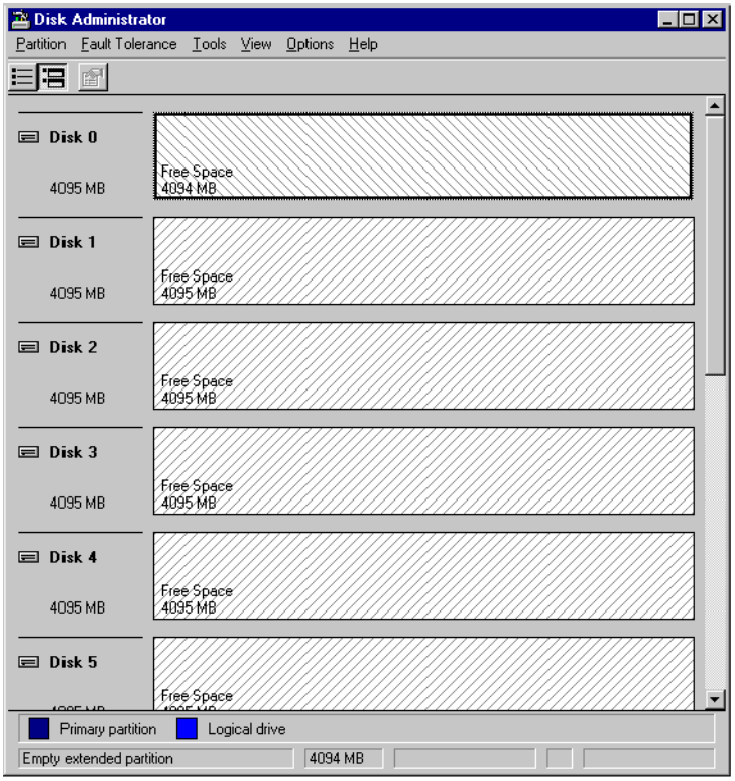
1. Choose Start > Programs > Administrative Tools (Common) > (Disk Administrator.

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

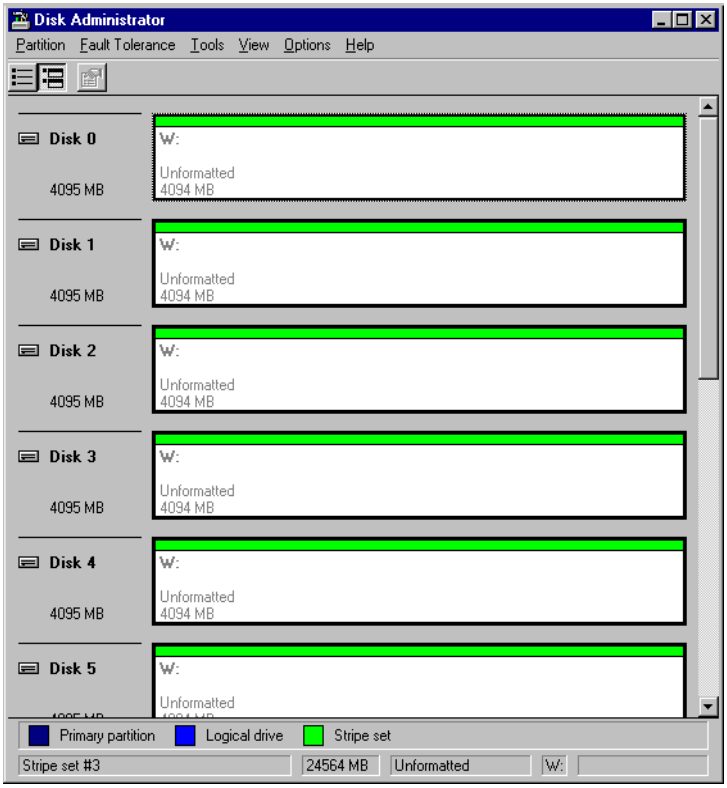


4. Repeat steps 2 and 3 for each of the five remaining unpartitioned hard disks.
5. Choose the first (topmost) drive.
6. Press Ctrl while choosing the remaining five drives so that all six hard disks are selected.
7. Choose Create Stripe Set from the Partition menu.
8. Choose OK on the Logical Volume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

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

   The Disk Administrator automatically adjusts the 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 the volume changes.

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



10. Reboot 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 the 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 the physical device level), undo file I/O contention may result, lowering the access times for both types of files.

Span Windows NT Server Virtual Memory Paging File Across Physical Volumes

Even if the Oracle9*i* database is the only network application running on Windows NT Server, some virtual memory paging is likely to happen as the Windows NT Server memory manager attempts to move an application's seldom-used pages to disk in order 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 the 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 the 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

It is recommended that the total combined size of the page files be at least the same size as the physical amount of RAM on the computer. Configurations where the combined size is two to four times the size of the physical RAM are not uncommon. Although the goal is to minimize paging as much as possible, a situation in which the operating system runs out of or low on paging space is to be avoided at all costs. Adequate paging files spaced across physical disks spread out the I/O most efficiently, since the operating system spreads paging evenly across page files.

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

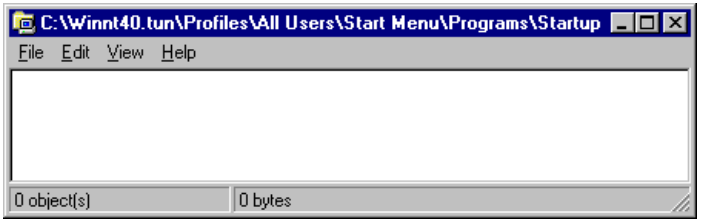
Close All Unnecessary Foreground Applications

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

- Startup Folder
- Virtual DOS Machines
- Screen Savers

Startup Folder

Remove applications from the Startup folders of Windows NT Server console operators. For example, if Microsoft Office Pro 95 is installed on the server, make sure that the FindFast indexing utility is not being loaded. The Office 95 Shortcut Toolbar is also unnecessary (using upwards of 3 MB when launched).



Virtual DOS Machines

Database administrators (DBAs) frequently run SQL scripts from MS-DOS 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

The 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.

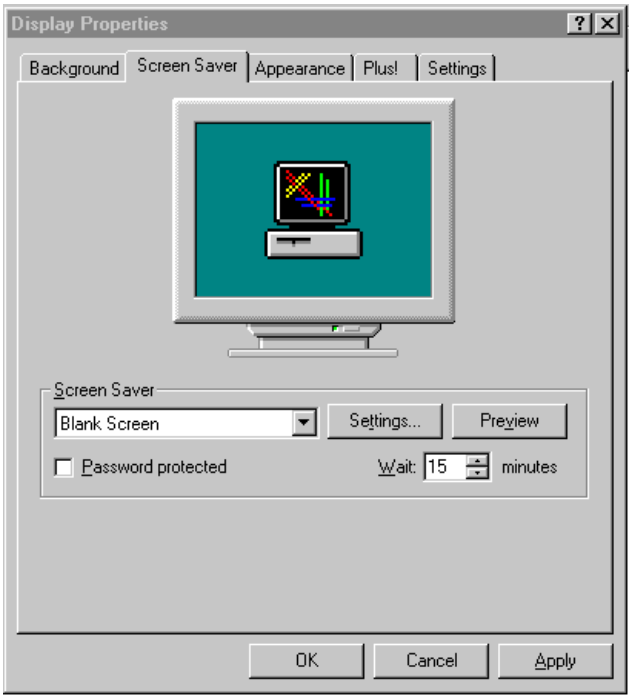
To set a blank screen saver:

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

   The Display Properties dialog box appears.

2. Choose the Screen Saver tab.

   The following information appears:



3. Select Blank Screen from the Screen Saver drop-down list box.
4. Choose 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 your Oracle9*i* database is one of the most critical operations that a database administrator (DBA) performs. For this reason, it is extremely important to choose the correct tools. Table 6–1 describes available backup and recovery tools and makes recommendations on which tool to use with your Oracle9*i* database:

Table 6–1 Backup and Recovery Tools

Tool	Description	Analysis of Tool	To Use this Tool, See...
Oracle Enterprise Manager Backup Management tools	The Oracle Enterprise Manager DBA Management Pack includes the 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. The 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 both online backups and the database is shut down.	Highly recommended for backing up and recovering your Oracle9*i* database because of capabilities and ease of use. Works in conjunction with Recovery Manager (RMAN). Backing up to tape requires an optional media management layer (MML) from a third-party vendor, such as Legato Storage Manager (LSM). LSM is included on your CD-ROM.	*Oracle Enterprise Manager Administrator's Guide*
Recovery Manager (RMAN) in command line mode[1]	Recovery Manager in command line mode is an Oracle9*i* tool that manages the process of 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. The user is unaware of this interface and instead interfaces only with a command line. [2]	Command line usage only. Must clearly understand the command line syntax described in *Oracle9i Recovery Manager Reference* before using. Backing up to tape requires an optional MML from a third-party vendor.	"Database Tools Overview" chapter of the *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.

Table 6–1 Backup and Recovery Tools

Tool	Description	Analysis of Tool	To Use this Tool, See...
Third-party vendor products	Third-party vendors such as Legato provide tape backup software that works with Backup Wizard and Recovery Manager.	If using Legato MML, you can back up to tape only if you are using RMAN and only if the tape device and the target database are on the same physical device. Contact your third-party vendors for additional information.	*Legato Storage Manager Administrator's Guide* and other third-party vendor documentation.
OCOPY utility	The OCOPY utility is one of several you may use when performing a user-managed backup and recovery. You can also make the following types of backups: ■ Backups when the 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	Command line usage only. Can back up only to disk. Can only perform local backups.	"Backing Up Files with the OCOPY Utility" and "Recovering Files with the OCOPY Utility" in this chapter and the *Oracle9i User-Managed Backup and Recovery Guide*

[1] On Windows NT, starting with 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* guide.

[2] 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.

Note: Before performing a backup when the database is shut down, and regardless of the backup tool you use (for example, Oracle Enterprise Manager Backup Wizard or a third party tool), stop the `OracleService`*`sid`* service in order to clear locks on the database files. If you do not stop this service, some database files may not be backed up.

Integrating RMAN with a Media Manager on Windows NT

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

Note: You do not need 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:** The `orasbt.dll` file does not have to be in the *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\bin` folder 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 Files with the OCOPY Utility

Use the OCOPY utility for both hot and raw database file backups of the file types described in Table 6–2:

Table 6–2 File Types for Database File Backups

File	Description
File Allocation Table (FAT) or NT File System (NTFS) files	FAT or NTFS files you want to back up.
Logical raw files: **Note:** For more information on logical drives, see Appendix A, "Storing Tablespaces on Raw Partitions"	A logical drive (accessed with the direct I/O) identified by the name: `\\.\`*`x`*`:` where *x*: is the logical drive designator.
Physical raw files	A physical hard drive (accessed with direct I/O) identified by a device name of the form: `\\.\physicaldrive`*`n`* where *n* is 0, 1, 2, and so on, representing each of the physical drives in the system.

Table 6–3 describes the two copy modes in which to use the OCOPY utility:

Table 6–3 The OCOPY Utility Copy Modes

Copy Mode	The OCOPY utility is used to...
Hard disk	Copy hot files to a new name and location on a hard disk where you can use an archive utility to back them up.

Table 6–3 The OCOPY Utility Copy Modes

Copy Mode	The OCOPY utility is used to...
Multiple diskettes	Back up large hot files directly to multiple diskettes.

Table 6–4 describes the syntax to use when backing up all file types to hard disk or multiple diskettes:

Table 6–4 Backing Up File Syntax

Copy Mode	File Type	Syntax
Hard disk	FAT or NTFS files	`C:\> ocopy old_file new_file`
	Logical raw	`C:\> ocopy \\.\C: new_file`
	Physical raw	`C:\> ocopy \\.\physicaldriven new_file`
Multiple diskettes	FAT or NTFS files	`C:\> ocopy /B hot_file a:`
	Logical raw	`C:\> ocopy /B \\.\C: a:`
	Physical raw	`C:\> ocopy /B \\.\physicaldriven a:`

Where...	Indicates the...
old_file	Name and location of the hot file you want to back up.
new_file	Name and location of the backup copy.
C:	Raw drive that holds a single RAW database file.
physical drive	Physical drive that holds a single RAW database file.
n	Number(s) representing each of the physical drives in the system.
hot_file	Path and filename of the hot file you want to back up.
a:	Diskette drive containing the diskette on which to save the backup copy. If the file is too large to fit on one diskette, the OCOPY utility prompts you to insert new diskettes as needed. The OCOPY utility catalogs the parts of the file automatically so that it can be reconstructed at a later time.
/B	Large files must be split over multiple diskettes.

Note: Always use a fresh diskette for each OCOPY utility procedure; do not use the OCOPY utility to back up a file onto a diskette that contains part or all of another file backed up using the OCOPY utility.

Recovering Files with the OCOPY Utility

Table 6–5 describes the syntax to use when recovering all file types from hard disk or multiple diskettes.

Table 6–5 Recovering File Syntax

Restore From...	File Type to Restore	Syntax
Hard Disk[1]	FAT or NTFS files	Use the Windows NT copy command.
	Logical raw	`C:\> ocopy new_file \\.\c:`
	Physical raw	`C:\> ocopy new_file \\.\physicaldrive`*n*
Multiple diskettes[2]	FAT or NTFS files	`C:\> ocopy /R a: restore_dir` **Note:** First insert the diskette containing the initial part of the backed up file.
	Logical raw	`C:\> ocopy /R a: \\.\c:`
	Physical raw	`C:\> ocopy /R a: \\.\physicaldrive`*n*

[1] These are files originally backed up without the /B option.

[2] These are files originally backed up with the /B option.

Where...	Indicates the...
`C:`	Raw drive, which holds a single RAW database file. If you use a different drive as your raw drive, substitute it for C:. The OCOPY utility prompts you to insert diskettes as needed.
new_file	File name to which to restore the file.
physical drive	Physical drive that holds a single RAW database file.
n	Number(s) representing each of the physical drives in the system.
/R	Restore option.
a:	Drive containing the diskette with the backed up file.
restore_dir	Directory on the server in which to place the file. (The restored file has the same name as the original file.)

Using Legato Storage Manager

When running backups to tape drivers using Legato Storage Manager (LSM), you must add LSM's `bin` directory to the PATH. The default location of the directory is `C:\win32app\nsr\bin`.

To add this directory to the PATH:

1. Select System from the Control Panel.

   The System Properties dialog box appears.

2. Select the Environment tab.

   The Environment view appears.

3. Select Path from the System Variables box.

4. Add the value, and choose Set.

5. Choose OK to close the System Properties dialog box.

Settings for Multiple Tape Devices

If you use LSM, and have more than one tape device on your system (LSM supports a maximum of four), ensure that the following settings are made in the LSM Administrator GUI:

1. Select Start > Programs > NetWorker group > NetWorker Administrator to start the LSM Administrator GUI.

2. Choose the set up server button, and set the parallelism field to the number of tape devices you have. Then, choose the devices button. For each tape device listed, select it with the right mouse button, select edit, and set the target sessions field to 1.

Japanese Version of Legato Storage Manager

In the Japanese version of Legato Storage Manager 5.5, the following log files are written in UTF8 format:

`C:\win32app\nsr\applogs\dmo.messages`

`C:\win32app\nsr\logs\daemon.log`

`C:\win32app\nsr\logs\messages`

On a Japanese version of Windows NT running Japanese Legato Storage Manager 5.5, you must run the following command at the command prompt to read one of these UTF8 log files:

```
C:\> nsrcat -N < file_name
```

where *file_name* is the appropriate path name for the specific log file, such as:

```
C:\win32app\nsr\applogs\dmo.messages
```

7

Oracle9*i* Database Specifications for Windows

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

This chapter contains these topics:

- Initialization Parameter File (init.ora) Overview
- Initialization Parameters Without Windows NT-Specific Values
- Calculating Database Limits

Initialization Parameter File (init.ora) Overview

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

- The amount of memory the database uses
- Whether to archive filled online redo logs
- Which control files currently exist for the database

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.

See Also:

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

Location of the Initialization Parameter File

By default, Oracle9*i* uses the initialization parameter files located in *`ORACLE_HOME`*`\admin\`*`db_name`*`\pfile\init.ora`, unless you specify a different initialization file with the `PFILE` option at database startup.

Editing the Initialization Parameter File

To customize Oracle9*i* database functions, you may need to edit the initialization parameter file. Only use an ASCII text editor to modify the file.

Sample File

A sample file called `initsmpl.ora` is located in the *`ORACLE_BASE`*`\`*`ORACLE_HOME`*`\admin\sample\pfile` directory.

> **Note:** If you create a database manually using a SQL script, you need to create an `init.ora` file or copy an existing `init.ora` file and modify the contents. If you use Oracle Database Configuration Assistant to create a database, the `init.ora` file is automatically created for you.

If you want to use the sample `initsmpl.ora` file 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 `init.ora` 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 the initialization parameters. These values and the annotations are preceded by comment signs (#), which prevent them from being processed. To activate a particular parameter, remove the preceding # sign. When you no longer want to use 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 small, medium, or large System Global Areas (SGAs), 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

The SGA_MAX_SIZE parameter holds the maximum size that the System Global Area (SGA) component can reach for a particular instance. With this release, Oracle9*i* can change its SGA configuration while the instance is running. This allows the sizes of the buffer cache, the shared pool, and the large pool to be changed without shutting down the instance.

Oracle9*i* can start instances underconfigured and allow the instance to use more memory by growing the SGA components, 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 init.ora is less than the sum of all components specified or defaulted to at initialization time, then the setting of SGA_MAX_SIZE in init.ora is ignored.

See Also:

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

The 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 the default values for many initialization parameters as being operating system-specific. However, not all the parameters that *Oracle9i Database Reference* describes as having operating system-specific values affect Windows NT. In these cases, Windows NT uses either the 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

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 the SGA in the core file for Oracle background processes.
SHADOW_CORE_DUMP	Specifies whether Oracle includes the SGA in the core file for foreground (client) processes.
CORE_DUMP_DEST	Specifies the directory where Oracle dumps core files.
CPU_COUNT	Oracle9*i* automatically sets value to number of CPUs 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 the Oracle 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 *ORACLE_BASE**ORACLE_HOME*\admin*db_name*\pfile\init.ora file. To display all parameter values (whether set in the init.ora 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.

The value for this parameter, whether defined in the *ORACLE_BASE**ORACLE_HOME*\admin*db_name*\pfile\init.ora file or the Oracle9*i* kernel, displays on-screen.

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 Database Initialization Parameters

Parameter	Description
CHARACTER SET[1]	Specifies the database Globalization Support character set to use. This parameter can be set only when you create the database.
DB_BLOCK_SIZE	Specifies the size in bytes of standard Oracle database blocks.
DB_NAME	Specifies the name of the database to be created. The 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 the size guidelines in Table 7–3 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 per 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)
Maximum number of files per database (depends on block size):	
■ 2 K sized blocks	20,000 files
■ 4 K sized blocks	40,000 files
■ 8 K sized blocks	65,536 files
■ 16 K sized blocks	65,536 files
Maximum file size for a FAT file	4 GB

Table 7–3 Block Size Guidelines

Type	Size
Maximum file size in NTFS	16 Exabytes (EB)
Maximum database size	65,536 * 64 GB equals approximately 4 Petabytes (PB)
Maximum number of extents per database (depends on block size). Typical values are:	
■ 2 KB sized blocks	121 extents
■ 4 KB sized blocks	255 extents
■ 8 KB sized blocks	504 extents
■ 16 KB sized blocks	1032 extents
■ 32 KB sized blocks	2070 extents
Shadow Process Memory:	
■ Release 9.0.1	444k
■ Release 8.1.7	335 K
■ Release 8.1.6	335 K
■ Release 8.1.3	265 K
■ Release 8.0.5	254 K
■ Release 8.0.4	254 K

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 Installation 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. The Windows NT Disk Administrator application enables you to create an extended partition on a physical drive.

An extended partition points to raw space on the disk that can be assigned multiple logical partitions for the 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 Oracle database server. Logical partitions can then be given symbolic link names to free up drive letters.

Disk Definition

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

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

where `Harddisk`*`m`* is the number of the physical drive, and `Partition`*`n`* is a logical partition number. `Harddisk`*`m`* starts at 0, and `Partition`*`n`* starts at 1.

`Partition0` has a special meaning in that it has access to the whole disk. For example, the first logical partition (E:) on the second physical drive in the above figure has the following entry:

```
\Device\Harddisk1\Partition1
```

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

```
\Device\Harddisk0\Partition1
```

Raw Partition Definition

Raw partitions are of two types:

- Physical Disk
- Logical Partition

> **Note:** Although you can use physical disks, Oracle Corporation recommends 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 the Disk Administrator. The *x* matches the *x* in `\Device\Harddiskx\Partition0`.

`\\.\PhysicalDrive`*x* is automatically defined by Windows NT 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 NT, the Disk Administrator writes a signature on the first block of all disks. To avoid overwriting that block, Oracle skips the first block of a physical raw partition that is used for an Oracle datafile.

Logical Partition

A logical partition is a partition created by the Disk Administrator that points to a drive other than `\Device\Harddisk`*x*`\Partition0`.

Logical partitions are initially assigned names with drive letters (`\\.\`*drive_letter*:) and typically re-assigned symbolic link names (`\\.\`*symbolic link name*). For example, `\\.\D`: may be assigned a symbolic link name of `\\.\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 the Disk Administrator. Symbolic link names can, on the other hand, be assigned using a utility such as `DOSDEV.EXE`, which is available with the Windows NT Resource Kit.

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

Physical Disk and Logical Partition Considerations

Consider the following when deciding which raw partition to use:

- Physical disks are automatically defined by Windows NT to represent the entire disk, and should never be defined by the user.
- Logical partitions 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 Oracle datafile and using a partition that occupies the entire disk for an Oracle datafile is not the same thing. Even when a partition occupies the entire disk, there is still a small space on the disk that is not part of the partition.

- If you are using an entire disk for an Oracle 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 the Disk Administrator, define and use a symbolic link name rather than a logical partition number (even if it occupies the entire disk).

Frequently Asked Questions

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

```
\\.\PhysicalDriveACCOUNTING_1 = \Device\Harddisk2\Partition1
\\.\PhysicalDriveACCOUNTING_2 = \Device\Harddisk3\Partition1
```

Answer: The Oracle database handles the datafile 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 the physical disk naming conventions. Oracle Corporation recommends that you convert to the logical partition at your earliest convenience. See "Compatibility Issues" on page A-6.

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 the 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-6 for information on rebuilding your Oracle database with the proper conventions.

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

Answer: Use the Oracle-provided OCOPY utility to copy data to/from a raw partition for both physical and logical raw conventions. See "Compatibility Issues" on page A-6 for further information.

Compatibility Issues

The 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 the OCOPY utility to copy from a physical disk to a logical partition, because the contents of these partitions are incompatible.

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

1. Perform a full database export to a (local) file system.
2. Create logical partitions and define logical names for these partitions.
3. 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.

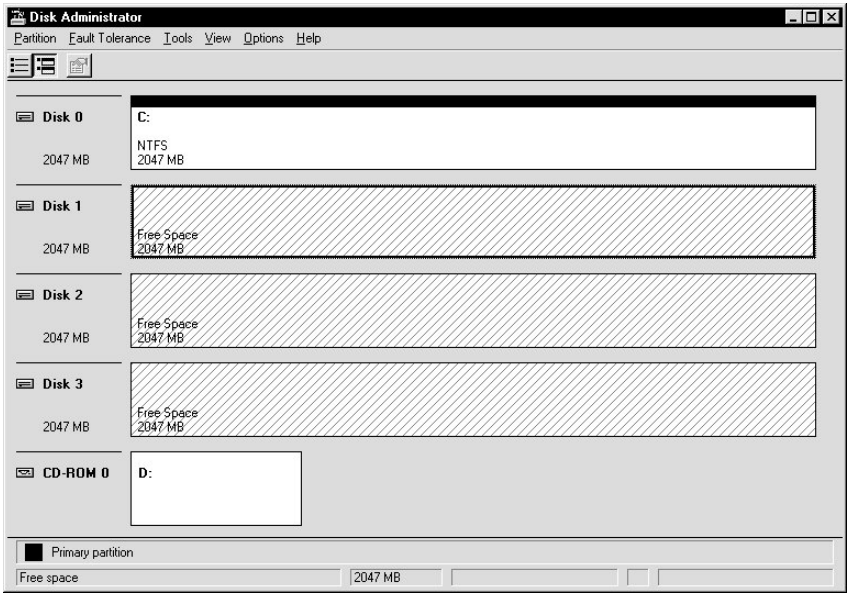
Creating an Extended Partition

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

To create an extended partition:

1. Choose Start > Programs > Administrative Tools > Disk Administrator.

   The Disk Administrator window appears.



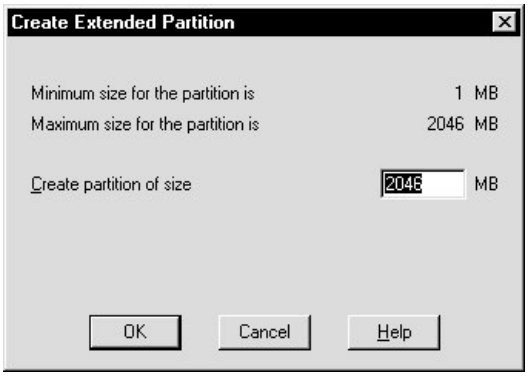
   Note that the lines display diagonally from top right to bottom left, indicating unpartitioned devices.

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 the 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.

To create logical partitions in an extended partition:

> **Note:** Oracle Corporation recommends you do not create more than 120 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.

   The Disk Administrator window displays the 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 you want your datafiles to 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. Choose Yes.

   A dialog box appears, informing you that the disks have been updated successfully.
7. Choose 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 Oracle repository database:

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

   where *password* is MANAGER for the SYSTEM user account by default. If you have changed this password, substitute MANAGER with the correct password.

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.

authorization

Permission given to a user, program, or process to access an Oracle database or operating system.

backup

A representative copy of data. This copy includes important parts of your database such as the control file, 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 the data available again. A backup is also a safeguard against an application error; if an application makes incorrect changes, you can restore the backup.

connect string

See net service name.

control file

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

data dictionary

A set of read-only tables that provide information about a database.

database alias

See net service name.

Dynamic Link Library (DLL)

An executable file that a Windows application can load when needed.

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

The Oracle architecture that ensures that database utilities, error messages, sort order, date, time, monetary, numeric, and calendar conventions automatically adapt to the 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 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 the type of computer), Oracle allocates a memory area called the System Global Area (SGA) and starts one or more Oracle processes. This combination of the SGA and the Oracle processes is called an instance. The memory and processes of an instance manage the associated database's data efficiently and serve the one or more users of the database.

instantiate, instantiation

Producing a more defined version of some object by replacing variables with values (or other variables).

In object-oriented programming, producing a particular object from its class template. This involves allocation of a structure with the types specified by the template, and initialization of instance variables with either default values or those provided by the constructor function of the class.

Internet Inter-ORB Protocol (IIOP)

A standard that enables Object Request Brokers (ORBs) from different vendors to communicate with each other using TCP/IP.

listener

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

listener.ora

A configuration file that describes one or more Transparent Network Substrate (TNS) listeners on a server.

Microsoft Management Console

An application that serves as a host for administrative tools called snap-ins. By itself, Microsoft Management Console does not provide any functionality.

migrate

To transform an installed version of an Oracle database from a major release to another major release, for example, from Oracle8 to Oracle9*i*. See "upgrade".

mount

To associate a database with an instance that has been started.

multiple Oracle homes

The capability of having more than one Oracle home on a computer.

net service name

The 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.

network service

In an Oracle application network, a service performs tasks for its service consumers. For example, a Names Server provides name resolution services for clients.

NT global groups

Contains users with access to computers and resources throughout the current domain and within any other domains that trust it. Global groups only contain global domain user accounts as their members.

Object Request Broker (ORB)

A software component that serves as the middleware between distributed objects. The distributed objects must comply with the Common Object Request Broker Architecture (CORBA) standard.

Optimal Flexible Architecture (OFA)

A set of file naming and placement guidelines for Oracle software and databases.

Oracle9*i* Enterprise Edition, Oracle9*i* Standard Edition and Oracle9*i* Personal Edition

The information in this guide applies to the Oracle9*i* Enterprise Edition, the Oracle9*i* Standard Edition, and the Oracle9*i* Personal Edition. Unless otherwise noted, the features and functionality described in this guide are common to both Oracle9*i* Enterprise Edition and Oracle9*i*.

ORACLE_HOME

Corresponds to the environment in which Oracle products run. This environment includes the location of installed product files, the PATH variable pointing to the products' binary files, registry entries, net service names, 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

The Oracle network interface that enables Oracle tools running on network workstations and servers to access, modify, share, and store data on other servers.

Oracle Protocol Support

A product that maps the functions of a given network protocol into Oracle Transparent Network Substrate (TNS) architecture. This process translates TNS function calls into requests to the underlying network protocol. This allows TNS to act as an interface among all protocols. Oracle Net requires Oracle protocol support.

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.

precompiler

A programming tool that enables you to embed SQL statements in a high-level source program.

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.

quota

A limit on a resource, such as a limit on the amount of database storage used by a database user. A database administrator can set tablespace quotas for each Oracle user name.

raw partition

A portion of a physical disk that is 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

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

redo log buffer

A circular buffer in the 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

The process of 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 NT registry and administered by Windows NT. Once a service is created and started, it can run even when no user is logged on to the computer.

service name

See net service name.

SID

See System Identifier (SID).

snap-in

An administrative tool that runs within Microsoft Management Console.

snapshot

(1) Information stored in rollback segments to provide transaction recovery and read consistency. Rollback segment information can be used to recreate 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.

synonym

An alias for a table, view, sequence, or program unit. A synonym is not actually an object itself; rather, it is a direct reference to its base object.

SYSDBA

A special database administration role that contains all system privileges with the ADMIN OPTION, and the 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 the 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 desired SID. The SID is included in the CONNECT DATA parts of the connect descriptors in a tnsnames.ora file, and in the definition of the network listener in a listener.ora file.

SYSTEM user name

One of two standard DBA user names automatically created with each database. (The other user name is SYS.) SYSTEM is created with an initial password of MANAGER. The SYSTEM user name is the preferred user name 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 applications and instances.

upgrade

To transform an installed version of an Oracle database major release into another major release of the same version. For example release 8.1.6 to release 8.1.7. See migrate.

username

A name that can connect to and access objects in a database.

view

A selective presentation of the structure of, and data in, one or more tables (or other views).

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W