Oracle® Database

2 Day DBA

10*g* Release 2 (10.2)

B14196-02

December 2005

Oracle Database 2 Day DBA, 10*g* Release 2 (10.2)

B14196-02

Primary Author: Colin McGregor

Contributing Author: Sushil Kumar, Antonio Romero, David Austin, Steve Fogel, Lance Ashdown, and Douglas Williams

Contributor: Michele Cyran, Daniela Hansell, Vasudha Krishnaswamy, Peter Laquerre, Venkat Maddali, Mughees Minhas, JP Polk, Mark Townsend, and Wanli Yang

Contents

Preface xi

Audience xi
Documentation Accessibility xi
Structure xii
Related Documentation xiii
Conventions xiv

1 Introduction

About this Book 1-1
What this Book is Not 1-1
How to Use this Book with Related Material 1-1
The Oracle Database 1-2
Common Oracle DBA Tasks 1-2
Tools for Administering the Database 1-2

2 Installing Oracle and Building the Database

Overview of Installation 2-1
Checking Prerequisites 2-1
Installation Choices 2-2
Basic Installation 2-2
Advanced Installation 2-3
Installing Oracle Software and Database 2-4
Using DBCA to Create and Configure a Database 2-6
Starting DBCA 2-7
Creating a Database with DBCA 2-7
Database Templates 2-8
Database Identification 2-8
Management Options 2-8
Database Credentials 2-8
Storage Options 2-8
Database File Locations 2-9
Recovery Configuration 2-9
Database Content 2-9
Initialization Parameters 2-10
Database Storage 2-11

Database Creation Options 2-11
Configuring Database Options with DBCA 2-12
Deleting a Database with DBCA 2-12
Managing Templates with DBCA 2-12
Advantages of Using Templates 2-12
Types of Templates 2-12
DBCA Templates Provided by Oracle 2-13
Creating Templates Using DBCA 2-13
Deleting DBCA Templates 2-14
Configuring Automatic Storage Management with DBCA 2-14
Upgrading a Database 2-14
Overview of the Database Upgrade Assistant 2-14
Pre-Upgrade Checks 2-15
Automated Upgrade Tasks 2-15
Support for Real Application Clusters 2-15
Support for Automatic Storage Management 2-15
Support for Silent Mode 2-15
Oracle Database Version Restrictions 2-16
Starting the Database Upgrade Assistant 2-16
Upgrading the Database Using the Database Upgrade Assistant 2-17
Installation: Oracle by Example Series 2-21

3 Getting Started with Oracle Enterprise Manager

Introduction to the Oracle Enterprise Manager Database Control 3-1
Database Management Features 3-1
Online Help 3-2
Navigational Features 3-2
Starting and Stopping the Oracle Enterprise Manager Console 3-2
Starting dbconsole in Windows 3-3
Accessing the Oracle Enterprise Manager Database Control 3-3
Granting Database Control Administrative Privileges 3-4
Setting Up Enterprise Manager Preferences 3-5
Defining Blackout Periods 3-5
Setting Preferred Credentials 3-6
SQL Statements and Other Management Tools: SQL*Plus and *i*SQL*Plus 3-6
Starting SQL*Plus 3-7
Starting *i*SQL*Plus 3-7
Managing Your Database: A Roadmap 3-7
Getting Started: Oracle by Example Series 3-8

4 Configuring the Network Environment

Understanding Network Configuration 4-1
Oracle Net Listener Configuration 4-1
Client Configuration 4-2
Connection Requests 4-2
Naming Methods 4-2
Local Naming 4-3

Directory Naming 4-3
Easy Connect Naming 4-3
External Naming 4-3
Tools for Network Configuration 4-3
Oracle Net Configuration Assistant 4-3
Enterprise Manager 4-4
Oracle Net Manager 4-4
Viewing Listener Configuration 4-4
Starting and Shutting Down the Listener 4-5
Configuring Oracle Networking on Client Machines 4-5
Networking: Oracle by Example Series 4-6

5 Managing the Oracle Instance

Overview of an Instance and Instance Management 5-1
About Initialization Parameters 5-2
Instance Memory Structure 5-2
The System Global Area (SGA) 5-2
Program Global Area (PGA) 5-3
Oracle Background Processes 5-3
Accessing the Database 5-4
About Database Administration Privileges 5-4
Server and Client Processes 5-5
Network Connections 5-5
Shutting Down and Restarting the Instance and Database with Enterprise Manager 5-6
Starting and Shutting Down the Database Instance on Windows 5-6
Viewing and Modifying Initialization Parameters 5-7
Managing Memory Parameters 5-9
Modifying Memory Parameters 5-9
Instances: Oracle by Example Series 5-9

6 Managing Database Storage Structures

Exploring the Storage Structure of Your Database 6-1
Control Files 6-2
Displaying Control File Information 6-2
Online Redo Log Files 6-2
Displaying Redo Log File Information 6-3
Multiplexing the Redo Log 6-3
Archive Log Files 6-4
Displaying Archive Log File Information 6-4
Rollback Segments 6-4
Displaying Rollback Segment Information 6-4
Datafiles 6-4
Viewing Datafile Information 6-5
Tablespaces 6-5
Locally-Managed vs. Dictionary-Managed Tablespaces 6-5
Tablespace Type 6-5

Tablespace Status 6-6
Auto-Extend Tablespace 6-6
Some Tablespaces in the Database 6-6
Other Storage Structures 6-7
Initialization Parameter File 6-7
Password File 6-8
Backup Files 6-8
Performing Common Database Storage Tasks 6-8
Viewing Tablespaces in Your Database 6-8
Creating a Tablespace 6-8
Modifying a Tablespace 6-9
Dropping a Tablespace 6-10
Reclaiming Wasted Space 6-10
Viewing Segment Advisor Recommendations 6-11
Running the Segment Advisor Manually 6-14
Managing Undo for Your Database 6-14
About Undo Data 6-14
Undo Tablespace Size and Retention Time 6-15
Automatic Undo Management 6-15
Managing Undo with Enterprise Manager 6-16
Using the Undo Advisor 6-17
Gaining Advice from Undo Advisor 6-18
Extending the Undo Tablespace 6-19
Setting Minimum Undo Retention Time 6-20
Making Changes to the Database Storage Structure 6-20
Storage: Oracle by Example Series 6-20

7 Administering Users and Security

Overview of Users and Security 7-1
Sample Schemas 7-2
Administering Roles 7-3
About Privileges and Roles 7-3
About Administrative Accounts 7-4
SYS 7-4
SYSTEM 7-5
Administrative Privileges 7-5
Viewing Roles 7-5
Creating Roles 7-6
Modifying Roles 7-6
Dropping Roles 7-7
Administering Profiles 7-7
Administering Database Users 7-8
Creating Users 7-8
Using a Shortcut to Create Users 7-10
Editing Users 7-10
Changing the Tablespace Quota for a User 7-10
Unlocking Accounts and Resetting Passwords 7-11

Granting Roles 7-11
Revoking Roles 7-12
Dropping Users 7-12
Users: Oracle by Example Series 7-12

8 Managing Schema Objects

Overview of Schema Objects 8-1
Exploring Database Schema Objects 8-1
Naming Schema Objects 8-2
Managing Tables 8-2
About Tables 8-3
Types of Table 8-3
Column Attributes 8-3
Table-Level Constraints 8-5
Storage Attributes 8-5
Other Table Creation Considerations 8-6
Viewing Tables 8-6
Viewing Table Data 8-7
Creating Tables 8-8
Estimating the Size of New Tables 8-9
Creating a Table: Example 8-9
Modifying Table Attributes 8-11
Adding a Column 8-11
Dropping a Column 8-11
Creating a New Table Constraint 8-11
Modifying an Existing Constraint 8-12
Dropping a Constraint 8-13
Loading Data Into Tables 8-13
Dropping a Table 8-15
Managing Indexes 8-15
About Indexes 8-16
Indexes and Performance 8-16
Index Types 8-16
Viewing Indexes 8-17
Creating Indexes 8-18
Dropping Indexes 8-18
Managing Views 8-19
About Views 8-19
Displaying Views 8-20
Creating a View: Example 8-20
Dropping Views 8-21
Managing Database-Resident Program Units 8-21
Managing Packages 8-22
Viewing Packages 8-22
Creating Packages 8-23
Editing Packages 8-23
Dropping Packages 8-24

Managing Package Bodies 8-24
Viewing Package Bodies 8-24
Managing Standalone Subprograms 8-25
Managing Triggers 8-25
Working with Other Schema Objects 8-26
Schemas: Oracle by Example Series 8-26

9 Performing Backup and Recovery

Overview of Database Backup and Recovery 9-1
Oracle Backup, Restore, and Recovery Concepts 9-2
Consistent and Inconsistent Backups 9-2
Media Recovery 9-3
The Flash Recovery Area 9-3
The RMAN Repository 9-4
Configuring Your Database for Basic Backup and Recovery 9-4
Planning Space Usage and Location for the Flash Recovery Area 9-4
Retention Policy and the Flash Recovery Area 9-4
Sizing the Flash Recovery Area 9-5
Credentials for Performing Oracle Enterprise Manager Backup and Recovery 9-6
Preferred Credentials for Backup and Recovery 9-6
Configuring the Flash Recovery Area 9-6
Configuring ARCHIVELOG Mode For the Database 9-7
Configuring Backup Settings 9-8
Understanding Backup Device Settings for Disk 9-8
Configuring Backup Policy Settings 9-9
Backing Up Your Database 9-10
Database Backup Concepts 9-10
Full Backups of Datafiles 9-10
Incremental Backups of Datafiles 9-10
Incrementally Updated Backups: Rolling Forward Image Copies of Datafiles 9-11
Using Tags to Identify Backups 9-11
Performing and Scheduling Backups with Enterprise Manager 9-11
Performing a Whole Database Backup with Oracle Enterprise Manager 9-11
Using the Oracle-Suggested Backup Strategy 9-15
Scheduling Other Backup Tasks 9-16
Validating Backups and Testing Your Backup Strategy 9-16
Performing Restore and Recovery Operations 9-17
Recovering a Whole Database from Backup 9-18
Recovering from a Lost SPFILE or Control File 9-20
Validating the Restore of Datafiles from RMAN Backup 9-20
Returning a Table to a Past State: Flashback Table 9-21
Enabling Row Movement on a Table 9-21
Performing Flashback Table 9-22
Recovering Dropped Tables: Flashback Drop 9-23
Managing Your Backups 9-24
Backup Management: Concepts 9-24
Using the Manage Current Backups Page 9-25

Searching for Backups on the Manage Current Backups Page 9-27
Managing Current Backups: Backup Sets 9-27
Managing Current Backups: Image Copies 9-27
Validating the Contents of Backup Sets or Image Copies........ 9-28
Crosschecking Backups 9-28
Deleting Expired Backups........ 9-29
Marking Backups as Available or Unavailable........ 9-30
Deleting Obsolete Backups 9-30
Displaying Backup Reports 9-30
Backup and Recovery: Oracle by Example Series........ 9-32

10 Monitoring and Tuning the Database

Proactive Database Monitoring........ 10-1
Alerts........ 10-1
Performance Self-Diagnostics: Automatic Database Diagnostics Monitor 10-2
Monitoring General Database State and Workload 10-2
Managing Alerts........ 10-4
Viewing Metrics and Thresholds........ 10-4
Setting Metric Thresholds........ 10-6
Responding to Alerts........ 10-7
Clearing Alerts 10-8
Setting up Direct Alert Notification 10-8
Diagnosing Performance Problems 10-9
Viewing Performance Analysis........ 10-9
Responding to Performance Findings........ 10-9
Modifying Default ADDM Behavior 10-10
Running ADDM Manually 10-10
Using Advisors to Optimize Database Performance........ 10-11
About Performance Advisors........ 10-11
Using the SQL Tuning Advisor........ 10-13
Using the SQL Access Advisor 10-14
Using the Memory Advisor........ 10-15
Setting New SGA or PGA Targets........ 10-15
Determining Performance Impact for Altering your SGA or PGA 10-16
Monitoring and Tuning: Oracle by Example Series 10-17

11 Managing Oracle Software

About Software Management and Patch Releases 11-1
Applying a Patch Release 11-1
Determining your Oracle Software Environment........ 11-1
Determining the Oracle Database Version........ 11-2
Determining the Location of your Oracle Home........ 11-2
Determining Your Hardware Configuration........ 11-3
Setting Your Oracle MetaLink Credentials 11-3
Staging the Patch Release........ 11-4
Installing the Patch Release 11-6

Managing Oracle Software: Oracle by Example Series 11-7

A Automatic Storage Management

What is Automatic Storage Management? A-1
Overview of Disks, Disk Groups, and Failure Groups A-2
Installing Automatic Storage Management A-3
Accessing the Automatic Storage Management Home Page A-3
Accessing the ASM Home Page on a Single Instance System A-4
Accessing the ASM Home Page On a Real Application Clusters System A-5
Starting Up and Shutting Down the ASM Instance A-5
Monitoring Disk Group Space Usage A-5
Creating a Disk Group A-6
Dropping a Disk Group A-8
Adding Disks to a Disk Group A-9
Dropping Disks from a Disk Group A-11
Backing Up ASM-Managed Files A-12
Automatic Storage Management: Oracle by Example Series A-12

B Overview of Real Application Clusters

What is Real Application Clusters? B-1
Real Application Clusters Database Configuration Scenario B-1
Differences in Administering Real Application Clusters and Single-Instance Oracle Databases B-2
Monitoring Oracle Real Application Clusters with Enterprise Manager B-2
Overview of the Cluster Database Home Page B-2
Viewing Real Application Clusters Database Targets B-4
Navigating to Oracle Enterprise Manager Cluster-Related Pages B-4
Viewing and Modifying Real Application Clusters Initialization Parameters B-5
Starting and Stopping Real Application Clusters Databases and Database Instances B-6
Starting and Stopping Real Application Clusters Databases B-6
Starting and Stopping Individual Instances in Real Application Clusters Databases B-6
Managing Real Application Clusters Backup and Recovery B-7
Redo Log Groups and Redo Threads in Real Application Cluster Databases B-7
Backing Up Real Application Clusters Databases B-8
Overview of Restore and Recovery Operations for a Real Application Clusters Database... B-8
Using a Flash Recovery Area with a Real Application Clusters Database B-9
Real Application Clusters: Oracle by Example Series B-9

Index

Preface

This preface contains these topics:

- Audience
- Documentation Accessibility
- Structure
- Related Documentation
- Conventions

Audience

Oracle Database 2 Day DBA is for anyone who wants to perform day-to-day administrative tasks with the Oracle Database. Prior knowledge or experience with managing databases is not required. The only requirement is a basic knowledge of computers.

In particular this book is targeted toward the following groups of Oracle users:

- Developers wanting to acquire basic DBA skills
- Anyone managing departmental servers
- Database administrators managing an Oracle database for small or medium business (SMB)

This book is equally useful for enterprise DBAs. It recommends best practices and describes efficient ways of performing administrative tasks with Oracle Enterprise Manager as the primary interface.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility 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 more information, visit the Oracle Accessibility Program Web site at

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

Accessibility of Code Examples in Documentation

Screen readers 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, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Structure

This document contains:

Chapter 1, "Introduction"

This chapter contains an brief overview of Oracle database administration.

Chapter 2, "Installing Oracle and Building the Database"

This chapter discusses how to install the Oracle software and database. It also describes how to configure additional databases, and how to upgrade and older version of a database to the current version.

Chapter 3, "Getting Started with Oracle Enterprise Manager"

This chapter introduces you to the Oracle Enterprise Manager and how to use it.

Chapter 4, "Configuring the Network Environment"

This chapter discusses how to configure a network so that clients can access your databases.

Chapter 5, "Managing the Oracle Instance"

This chapter discusses starting and stopping the database instance, and managing initialization parameters.

Chapter 6, "Managing Database Storage Structures"

This chapter discusses management of the database's storage structure.

Chapter 7, "Administering Users and Security"

This chapter discusses how to add and maintain user accounts. It includes information about using system privileges and roles to control user access to the database.

Chapter 8, "Managing Schema Objects"

This chapter discusses managing tables, indexes, and other schema objects.

Chapter 9, "Performing Backup and Recovery"

This chapter discusses how to backup and recover your database.

Chapter 10, "Monitoring and Tuning the Database"

This chapter discusses monitoring database activities and diagnosing performance problems.

Chapter 11, "Managing Oracle Software"

This chapter discusses how to keep your Oracle software up-to-date with product fixes.

Appendix A, "Automatic Storage Management"

This appendix introduces Automatic Storage Management.

Appendix B, "Overview of Real Application Clusters"

This appendix introduces Real Application Clusters.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Database Administrator's Guide*
- *Oracle Enterprise Manager Advanced Configuration*
- *Oracle Database Concepts*
- *Oracle Enterprise Manager Concepts*
- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database SQL Reference*
- *Oracle Database Reference*
- *Oracle Database Backup and Recovery Basics*
- *Oracle Database Backup and Recovery Advanced User's Guide*
- *Oracle Real Application Clusters Installation and Configuration Guide*
- *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Database Installation Guide*

Many of the examples in this book use the sample schemas, which are installed by default when you select the Basic Installation option with an Oracle Database installation. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Printed documentation is available for sale in the Oracle Store at

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

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

```
http://otn.oracle.com/membership/
```

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

```
http://otn.oracle.com/documentation/
```

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

As an Oracle Database Administrator (DBA), you are responsible for the overall operation of the Oracle database.

This chapter contains the following topics:

- About this Book
- The Oracle Database
- Common Oracle DBA Tasks
- Tools for Administering the Database

About this Book

Oracle Database 2 Day DBA is a database administration quick start guide that teaches you how to perform day-to-day database administrative tasks. The goal of this book is to help you understand the concepts behind the Oracle Database. It teaches you how to perform all common administration tasks needed to keep the database operational, including how to perform basic troubleshooting and performance monitoring activities.

The primary administrative interface used in this book is Oracle Enterprise Manager in Database Console mode, featuring all the self-management capabilities introduced in the Oracle Database.

What this Book is Not

Oracle Database 2 Day DBA is task oriented. The objective is to describe why and when administrative tasks need to be performed. Where appropriate, it describes the concepts necessary for understanding and completing the task at hand, assuming the reader has no prior knowledge of the database. This book is not an exhaustive discussion of all Oracle database concepts. For this type of information, refer to *Oracle Database Concepts*.

Additionally, for a complete discussion of administrative tasks, refer to *Oracle Database Administrator's Guide.*

How to Use this Book with Related Material

This book is part of comprehensive set of learning material for administering an Oracle Database, which includes a 2 Day DBA Oracle by Example (OBE) series, available on the Web, and in an Oracle University instructor-led class.

Every chapter in *Oracle Database 2 Day DBA* has an associated Oracle By Example lesson. The OBE steps through all tasks in the chapter and includes annotated screen shots. In some cases, the OBE provides additional information to help you complete the task at hand.

At the end of each chapter, you can find the link to its associated OBE lesson. The home page for the 2 Day DBA Oracle by Example series is as follows:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/index.html
```

In addition, Oracle University provides a related instructor-led course called Oracle Database 10g: 2 Day DBA. Contact Oracle University for more information.

The Oracle Database

Oracle is a relational database. In a relational database, all data is stored in two-dimensional tables that are composed of rows and columns. The Oracle Database enables you to store data, update it, and efficiently retrieve it.

Oracle provides software to create and manage the Oracle database. The database consists of physical and logical structures in which system, user, and control information is stored. The software that manages the database is called the Oracle database server. Collectively, the software that runs Oracle and the physical database are called the Oracle database system.

You will learn more about the operation of the database server and the structure of the Oracle database where they are relevant to the performance of specific database management tasks.

Common Oracle DBA Tasks

As an Oracle DBA, you can expect to be involved in the following tasks:

- Installing Oracle software
- Creating Oracle databases
- Performing upgrades of the database and software to new release levels
- Starting up and shutting down the database
- Managing the database’s storage structures
- Managing users and security
- Managing schema objects, such as tables, indexes, and views
- Making database backups and performing recovery when necessary
- Proactively monitoring the database’s health and taking preventive or corrective action as required
- Monitoring and tuning performance

In a small to midsize database environment, you might be the sole person performing these tasks. In large, enterprise environments, the job is often divided among several DBAs, each with their own specialty, such as database security or database tuning.

Tools for Administering the Database

The intent of this book is to allow you to quickly and efficiently create an Oracle database, and to provide guidance in basic database administration.

The following are some of the products, tools, and utilities you can use in achieving your goals as a database administrator:

- **Oracle Universal Installer (OUI)**

  The Oracle Universal Installer installs your Oracle software and options. It can automatically launch the Database Configuration Assistant to install a database.

- **Database Configuration Assistant (DBCA)**

  The Database Configuration Assistant creates a database from templates that are supplied by Oracle, or you can create your own. It enables you to copy a preconfigured seed database, thus saving the time and effort of generating and customizing a database from scratch.

- **Database Upgrade Assistant**

  This Database Upgrade Assistant guides you through the upgrade of your existing database to a new Oracle release.

- **Oracle Net Manager**

  Net Manager is an alternate tool for configuring and managing Oracle Database networks.

- **Oracle Enterprise Manager**

  The primary tool for managing your database is Oracle Enterprise Manager, a web-based interface. After you have installed the Oracle software, created or upgraded a database, and configured the network, you can use Oracle Enterprise Manager for managing your database. In addition, Oracle Enterprise Manager also provides an interface for performance advisors and for Oracle utilities such as SQL*Loader and Recovery Manager.

2

Installing Oracle and Building the Database

This chapter describes how to install Oracle software and a starter database with the Oracle Universal Installer (OUI). It also describes upgrading an older database to the current release.

This chapter contains the following topics:

- Overview of Installation
- Installing Oracle Software and Database
- Using DBCA to Create and Configure a Database
- Upgrading a Database
- Installation: Oracle by Example Series

Overview of Installation

To install your Oracle software you use the Oracle Universal Installer (OUI). The Oracle Universal Installer is a graphical user interface (GUI) tool that enables you to view the Oracle software that is already installed on your machine, install new Oracle software, and deinstall Oracle software that you no longer intend to use. Online Help is available to guide you through the installation process.

Checking Prerequisites

Before installing the software, the installer performs a number of automated checks to ensure that your machine fulfills the basic hardware and software requirements for an Oracle Database installation. If your machine does not meet a requirement, an error message is displayed. The requirements may vary depending upon the type of computer and operating system you are running, but some prerequisites include:

- A minimum of 512 MB of memory is available.
- Sufficient paging space is available.
- Appropriate operating system service packs or patches are installed.
- An appropriate file system format is being used.

The installer automatically sets any operating system environment variables that the Oracle database server requires for its operation.

> **See Also:** *Oracle Database Installation Guide* for your platform for more information about exact requirements

Installation Choices

The Oracle Universal Installer guides you through an interview phase that asks you to specify your choices for installation and database creation. The exact sequence of steps depends on your operating system.

The installation options are divided into basic and advanced. During a basic installation, you need only make basic choices. For all other cases, you make both basic and advanced choices.

Basic Installation

When you install Oracle Database during basic and advanced installations, you make the following choices.

- What database product do you want to install?

  You can install one of the following:

  - **Enterprise Edition**—Oracle's full featured database product which provides data management for enterprise-level applications. It is engineered for mission-critical, high-security online transaction processing (OLTP) and data warehousing environments.
  - **Standard Edition**—This installation type is suitable for workgroup or department-level applications, and for small to medium sized enterprises. It is engineered to provide core relational database management services and options.
  - **Personal Edition** (Windows operating systems only)—This installation type installs the same software as the Enterprise Edition, but supports only a single-user development and deployment environment.
  - **Custom Install**—This installation type enables you to customize the Enterprise Edition. Use Custom Install to add a component to your existing installation that would not normally be installed or prevent certain components from being installed.

- Do you want to create a starter database in addition to installing the Oracle software? To create the database, the installer automatically launches the Database Configuration Assistant at the end of the installation.

  To install a preconfigured database, select one of the following database types, optimized according to usage:

  - General Purpose
  - Transaction Processing
  - Data Warehouse

  To create an custom database in which you configure your own database structure, see advanced installation.

  > **Best Practice:** Oracle recommends that you install a preconfigured database, which is faster and easier to do. You can customize the database later.

- What are your Database Configuration Options? These options include your Global Database Name and Oracle System Identifier (SID).

The Global Database Name is the full name of the database that uniquely distinguishes it from any other database. The global database name is of the form `database_name.database_domain`, as in `sales.us.acme.com`. The database name portion `sales` is a simple name you call your database. The database domain portion `us.acme.com` specifies the database domain in which the database is located. Together, the database name and domain make up the Global Database Name.

During basic installation, Oracle automatically installs the sample schemas. Many of the examples in the database documentation are based on the sample schemas.

You will also be prompted to enter your database schema password for the `SYS` and `SYSTEM` accounts, which enable you to manage and administer the database.

Advanced Installation

During advanced installations, you are prompted to make the choices listed in this section, in addition to the basic options. The installation process provides defaults for every choice.

The advanced installation choices are described here. However, advanced installation steps are not documented in this chapter. For more information, see the *Oracle Database Installation Guide* for your platform.

During an advanced installation, you are given the following options:

- Install the sample schemas?

  To complete the exercises in this book and related course material, you must install the sample schemas. This data is also used in most examples throughout Oracle Database documentation. Oracle recommends that you install the sample schemas.

  This choice is a configuration option only during advanced installation. This option defaults to No during advanced installation. However, sample schemas are installed by default during basic installation.

- What kind of storage mechanism do you want your database to use?

  A database includes several files that store the user data, database metadata, and information required to recover from failures. As an administrator, you decide what kind of storage subsystem to use for these files. You can select from the following options:

  - **File System**—This default option creates database files that are managed by your operating system's file system. You can specify the directory path where database files are to be stored. Oracle Database can create and manage the actual files.

    If you are not certain of which option to use, choose the default selection of File System.

  - **Automatic Storage Management**—This functionality enables you to define a pool of storage (called a disk group) in which Oracle automatically manages database file placement and naming. For environments with a large number of disks, this option simplifies database administration and maximizes performance. Disk mirroring and striping are performed automatically.

    Automatic Storage Management requires a separate instance to configure and manage disks groups. The installer guides you through the creation and launching of an Automatic Storage Management instance and its components. For more information, see Appendix A, "Automatic Storage Management" and *Oracle Database Administrator's Guide*.

  - **Raw Devices**—This option enables you to manage your storage devices outside of the operating system file system by providing Oracle Database with unformatted physical disk space called raw devices. This method is primarily used in Oracle Real Application Clusters (RAC) environments. If you select this option, ensure that the raw devices have been created with an operating system command, and that they are not being used for any other purposes or by other Oracle databases.

- What are your Database Management Options?

  Here you specify whether to manage your database centrally or locally. The central deployment mode enables you to manage multiple targets, such as databases and application servers, from a single console. The local deployment mode can manage only a single database instance.

  The central deployment mode requires a special agent to be installed on the database host. If no such agent is found, then this option is disabled.

  If you are setting up a single database for the first time, it is recommended that you deploy Enterprise Manager locally, which is the default.You can later install additional databases and migrate your deployment mode to central management.

At the end of the interview, a summary window displays the options that you have chosen, the requirements for space based on those options, and the components that will be installed. If you have chosen to install a database, then this step is performed immediately after the software installation is complete.

Installing Oracle Software and Database

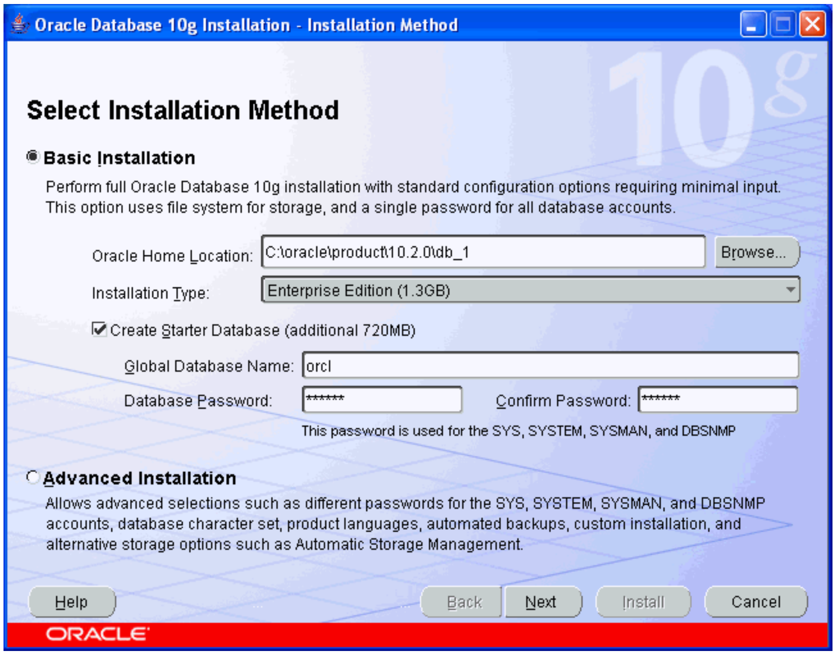
This section describes the basic installation steps. Most steps are common to all platforms and involve running the Oracle Universal Installer. Platform-specific steps are noted.

The following steps provide a summary of the installation process. For further assistance along the way, consult the online help or the *Oracle Database Installation Guide* for your platform.

1. Log on to your computer as a member of the administrative group that is authorized to install Oracle software and to create and run the database. Refer to your operating system-specific documentation or contact your system administrator to determine whether you have the necessary privileges to install new software.
2. Insert the distribution media for the database into your computer. The Autorun window opens automatically. Click **Install/Deinstall Products**.

   If downloading from Oracle's Web site, then follow the instructions on the site.
3. The Oracle Universal Installer Select Installation Method window appears. Figure 2–1 shows what this window looks like on Windows.

Figure 2–1 Oracle Universal Installer Select Installation Method Window



Select Installation Method

● **Basic Installation**

Perform full Oracle Database 10g installation with standard configuration options requiring minimal input. This option uses file system for storage, and a single password for all database accounts.

Oracle Home Location: C:\oracle\product\10.2.0\db_1 Browse...

Installation Type: Enterprise Edition (1.3GB)

☑ Create Starter Database (additional 720MB)

Global Database Name: orcl

Database Password: ****** Confirm Password: ******

This password is used for the SYS, SYSTEM, SYSMAN, and DBSNMP

○ **Advanced Installation**

Allows advanced selections such as different passwords for the SYS, SYSTEM, SYSMAN, and DBSNMP accounts, database character set, product languages, automated backups, custom installation, and alternative storage options such as Automatic Storage Management.

4. You must now decide which type of installation to perform:

- **Basic Installation**—Select this option to quickly install Oracle Database 10g. This method requires minimal user input. It installs the software and optionally creates a general-purpose database based on the information you provide.

  For basic installation, you specify the following:

  – **Oracle Home Location**—Enter the directory in which to install the Oracle Database 10g software. You must specify a new Oracle home directory for each new installation of Oracle Database 10g.

  – **Installation Type**—Select either **Enterprise Edition**, **Standard Edition**, or **Personal Edition (Windows Only)**. Refer to "Basic Installation" on page 2-2.

  – **UNIX DBA Group** (Linux and Unix only)—Specify the DBA group for your operating system.

  – **Create Starter Database** —Check this box to create a database during installation. Oracle recommends that you create a starter database for first-time installations. You must enter a **Global Database Name** and **Database Password** for the `SYS`, `SYSTEM`, `SYSMAN`, and `DBSNMP` administrator accounts. The Global Database Name is described in "Basic Installation".

  To start the basic installation, click **Next**. The Preparing to Install progress bar appears.

- **Advanced Installation**—Select this option to customize your installation. For example, use this method to install Oracle Real Application Clusters, to upgrade a database, to configure Automatic Storage Management, or to configure automated backups. Selecting this option steps you through installation steps not documented here. For more information about the advanced choices, see "Advanced Installation" in this chapter. Also see the *Oracle Database Installation Guide* for your platform.

5. The Product-Specific Prerequisite Checks window appears. Refer to "Checking Prerequisites" on page 2-1. The Installer performs a number of environment checks and indicates success, warning, or failure. Details of the checks are provided. The installation can only proceed with check status of Succeeded or Warning. If you have failed checks, you must resolve them manually.

   After verifying that your system meets the minimum requirements for installing and configuring the chosen products, click **Next**.

6. **Linux and Unix Only**. If this is the first time you are installing any Oracle software on this machine, then the following occurs:

   The Specify Inventory Directory window appears. You must specify a directory for installation files and the name of an operating system group that has write permission to the directory. OUI uses this location to keep track of all Oracle software. This information is used while applying patches or upgrading an existing installation, and while deinstalling Oracle software. Note that this area is different from the Oracle home, which you specify later.

   Click **Next** to continue.

7. A summary screen appears showing information such as your global settings, space requirements and the new products to be installed. Click **Install** to start the installation. The Install window appears showing installation progress.

8. At the end of the installation phase, the Configuration Assistants window appears. This window lists the configuration assistants that are started automatically. If you are creating a database, then the Database Configuration Assistant starts automatically in a separate window.

   At the end of database creation, you are prompted to unlock user accounts to make the accounts accessible. The `SYS` and `SYSTEM` accounts are already unlocked. Click **OK** to bypass password management.

9. **Linux and Unix Only**. The Execute Configuration scripts window appears. You are prompted to open a new terminal window, and to run scripts as the root user. After you run the scripts, return to this window and click **Ok**.

10. Your installation and database creation is now complete. The End of Installation window displays several important URLs, one of which is for Enterprise Manager. You can navigate to this URL in your browser and log in as the SYS user with the associated password, and connect as SYSDBA. You use Enterprise Manager to perform common database administration tasks. See Chapter 3, "Getting Started with Oracle Enterprise Manager".

11. After noting the information in the End of Installation screen, you can exit the Installer.

Using DBCA to Create and Configure a Database

During installation, you can direct the installer to create and configure a new database. If you have already done so, you can skip to the next chapter.

However, if you install Oracle software only, or if you want to create additional databases with the release software that you just installed, then you can use the Database Configuration Assistant (DBCA).

DBCA enables you to perform the following tasks:

- Creating a Database with DBCA
- Configuring Database Options with DBCA
- Deleting a Database with DBCA
- Managing Templates with DBCA
- Configuring Automatic Storage Management with DBCA

Online Help is available by clicking **Help**. It provides information that guides you in selecting configuration options.

Starting DBCA

To launch DBCA:

1. Log on to your computer as a member of the administrative group that is authorized to install Oracle software and create and run the database.
2. To launch DBCA on a Windows operating system, click **Start** and then select **Programs**, **Oracle -** ***home_name***, **Configuration and Migration Tools**, and then **Database Configuration Assistant**.

   To launch the DBCA on UNIX, or as another method on a Windows operating system, enter the following command at a system prompt:

   ```
   dbca
   ```

   The `dbca` utility is typically located in *`ORACLE_HOME`*`/bin`.

   The Welcome window appears.
3. Click **Next** to continue. The DBCA Operations window appears.

Creating a Database with DBCA

On the DBCA Operations window, select **Create a Database** to start a wizard that enables you to create and configure a database. The wizard requests your input on the following:

- Database Templates
- Database Identification
- Management Options
- Database Credentials
- Storage Options
- Database File Locations
- Recovery Configuration
- Database Content
- Initialization Parameters
- Database Storage
- Database Creation Options

Note that most windows of the wizard provide a default setting. To accept all the default parameters, you can click **Finish** at any step. DBCA displays a final confirmation window. Click **OK** to initiate the database creation.

Database Templates

This window enables you to select the type of database you want to create. By default, Oracle ships pre-defined templates. There are templates for **Data Warehouse**, **General Purpose**, and **Transaction Processing** databases.

The templates contain settings optimized for workload. Click **Show Details** to see the configuration for each type of database. Choose the template suited to the type of workload your database will support. If you are not sure which to choose, select the default General Purpose template.

For more complex environments, you can select the **Custom Database** option. This option results in a more extensive interview, which means that it will take longer to create your database because a database creation script must be run.

For more information about using database templates, see "Managing Templates with DBCA" on page 2-12.

Database Identification

In the **Global Database Name** box, enter the database name in the form *database_name.domain_name*.

In the **SID** box, enter the Oracle system identifier. The SID defaults to the database name and uniquely identifies the instance that runs the database.

Management Options

Use this window to set up your database so it can be managed with Oracle Enterprise Manager. Oracle Enterprise Manager provides Web-based management tools for individual databases, as well as central management tools for managing your entire Oracle environment.

Check **Configure the Database with Enterprise Manager to use Enterprise Manager.** Select one of the following options:

- If the Oracle Management Agent has been installed on your host computer, then you have the option of selecting central management by selecting **Use Grid Control for Database Management**. If you select this type of management, you must also indicate which management service to use in the drop-down menu.
- Select **Use Database Control for Database Management** to manage your database locally. If you choose this option, you can additionally check Enable Email Notifications for Oracle to e-mail you alerts regarding potential problems, and check Enable Daily Backup. Click Help for more information about these options.

Database Credentials

In this window, you specify the passwords for the administrative accounts such as `SYS` and `SYSTEM`. Select one of the following options:

- Select **Use the Same Password for All Accounts** and enter the password to use the same password for all accounts.
- Select **Use Different Passwords** and specify passwords individually.

Storage Options

Specify the type of storage mechanism you would like your database to use. For more information, refer to "Installation Choices" on page 2-2.

Database File Locations

In this window, you specify the Oracle home and directory path in which to install the Oracle software. Choose one of the following:

- **Use Database File Locations from Template**—Selecting this option instructs the DBCA to use the directory information as specified in the template. Later, you can make modifications to database filenames and locations.
- **Use Common Location for All Database Files**—This option requires you to specify a new directory for the Oracle home. All the database files will be created in this location. Later, you can make modifications to database filenames and locations.
- **Use Oracle-Managed Files**—Select this option to have Oracle directly manage operating system files comprising an Oracle database. You specify default location called a database area for all your files. Oracle thereafter automatically creates and deletes files in this location as required. You can also create multiple copies of your redo and online log files by selecting Multiplex Redo Logs and Control Files. To learn more about redo logs and control files, refer to Chapter 9, "Performing Backup and Recovery".

  Selecting this option enables you to delegate the complete management of database files to the database. You no longer need to specify the filenames, location, or their sizes.

Recovery Configuration

When you create a new database, it is important to configure the database so you can recover your data in the event of a system failure. You can select the following options:

- **Specify Flash Recovery Area**—Select this option to specify a backup and recovery area and specify its directory location and size. You can use variables to identify standard locations. To review or add additional locations, click **File Location Variables** at the bottom of the window.
- **Enable Archiving**—Select this option to enable archiving of database redo logs, which can be used to recover a database. Selecting this option is the same as enabling Archive Log Mode in Oracle Enterprise Manager or running the database in `ARCHIVELOG` mode. You can accept the default archive mode settings or change them by selecting **Edit Archive Mode Parameters**.

  Oracle recommends you select **Enable Archiving**. Selecting this option provides better protection for your database in the case of software or hardware failure. If you do not select this option now, you can enable archive log mode later. See "Configuring Your Database for Basic Backup and Recovery" on page 9-4.

Database Content

When you create your database, you can load it with data.

Sample Schemas

Check **Sample Schemas** to include the Sample Schemas (`EXAMPLE`) tablespace in your database. The Sample Schemas provide a common platform for examples. Oracle books and educational materials contain examples based upon the Sample Schemas. Oracle recommends they be included in your database.

Custom Scripts

In the **Custom Scripts** tab, you can specify one or more SQL scripts to run after your database is created. Scripts are useful for performing post-installation tasks, such as loading custom schemas. To specify scripts, check Run the following scripts; otherwise accept the default No scripts to run. Note that if you choose to run scripts after installation, your scripts must include a connect string that identifies the database. Click Help for more information.

Initialization Parameters

The links on this window provide access to windows that enable you change default initialization parameter settings. These parameters fall into the following categories:

- Memory
- Sizing
- Character Sets
- Connection Mode

Memory Use this window to set the initialization parameters that control how the database manages its memory usage. You can choose from one of the following approaches to memory management:

Typical—This method requires little configuration, and allocates memory as a percentage of total overall physical system memory. Select Typical and enter a percentage value. Click Show Memory Distribution to see how much memory the DBCA assigns to the System Global Area (SGA) and the Program Global Area (PGA). To learn more about PGA and SGA, refer to "Managing Memory Parameters" in Chapter 5, "Managing the Oracle Instance".

Custom—This method requires more configuration, but gives you more control over how the database uses system memory. This option is meant for more experienced database administrators. You can directly specify memory sizes for the SGA and PGA and their sub-structures, such as the shared pool and buffer cache.

Select one of the following options:

- Select **Automatic** to allocate specific amounts of memory to SGA and PGA.
- Select **Manual** to enter specific values for each SGA component. This will customize how the SGA memory is distributed among the SGA memory substructures.

Sizing In this tab, you specify the smallest block size and the maximum number of operating system user processes that can simultaneously connect to the database.

In the **Block Size** list, enter the size in bytes or accept the default. Oracle Database data is stored in these blocks. One data block corresponds to a specific number of bytes of physical space on disk. While using pre-defined templates, this field is not enabled since the database will be created with the default block size of 8 KB. But while using the custom option, you can change block size. Selecting a block size other than the default 8 KB value requires advanced knowledge and should only be done when absolutely required.

In the **Processes** field, specify the maximum number of processes that can simultaneously connect to the database. Enter a number or accept the default of 150. The default value for this parameter is good enough for many environments.The value should be 6 or greater. This value should allow for all background processes, such as locks and parallel execution processes.

Character Sets Use this window to define the character sets used by your database. Character sets are the encoding schemes used to display characters on your computer screen. Choosing a character set determines what languages can be represented in the database.

For **Database Character Set**, select from one of the following options:

- **Use the Default—**Select this option if you need to support only the language currently used by the operating system for all your database users and your database applications.
- **Use Unicode (AL32UTF8)—**Select this option if you need to support multiple languages for your database users and your database applications.
- **Choose from the list of character sets—**Select this option if you want the Oracle Database to use a character set other than the default character set used by the operating system.

In the **National Character Set** list, select a character set or accept the default. The national character set is an alternative character set that enables you to store Unicode characters in a database that does not have a Unicode database character set. Choosing a national character can make programming in the national character set easier.

In the **Default Language** list, select a default database language or accept the default. The default language determines how the database supports locale-sensitive information such as day and month abbreviations, default sorting sequence for character data, and writing direction (left or right).

In the **Default Date Format** list, select a date format or accept the default. The default date format determines the convention for displaying the hour, day, month, and year. For example, in the United Kingdom, the date format is DD-MM-YYYY.

Connection Mode Use this window to select the database mode. You can run the database in either of the following modes:

- **Dedicated Server Mode** allows a dedicated server process for each user process. Select this option when the number of total clients is expected to be small, or when database clients will make persistent, long-running requests to the database.
- **Shared Server Mode** allows several client connections to share a database-allocated pool of resources. Use this mode when a large number of users need to connect to the database. It is also useful when database memory is limited or when better performance is needed, because you can have more client connections to the database than in dedicated server mode. If you choose shared server mode, then you must also indicate the number of server processes you want to create when an instance is started. For more information about setting this parameter, click Help.

Database Storage

A navigation tree displays the storage structure of your database (control files, datafiles, redo log groups, and so forth). If you are not satisfied with the storage structure or parameters, then you can make changes. You can create a new object with **Create** and delete existing objects with **Delete**.

Note that if you selected one of the preconfigured templates for a database, then you cannot add or remove control files, datafiles, or undo segments.

Database Creation Options

Check any of the following options for creating the database:

- **Create Database—**Check to create your database at this time.

- **Save as a Database Template**—Check to save the database definition as a template to use at another time.
- **Generate Database Creation Scripts**—Check to generate a SQL database creation script that you can run at a later time.

Configuring Database Options with DBCA

In the Operations window, select **Configure Database Options** to change various aspects of your database configuration. For example, you can change the database from a dedicated server to a shared server. You can add database options that have not been previously configured for use with your database, for example, Oracle Label Security or Oracle OLAP.

Deleting a Database with DBCA

In the Operations window, select **Delete a Database** to remove a database from the system. When you select this option, DBCA deletes all the files associated with this database. On Windows, any associated services are also deleted.

Managing Templates with DBCA

DBCA templates are XML files that contain information required to create a database. Templates are used in DBCA to create new databases and clone existing databases. The information in templates includes database options, initialization parameters, and storage attributes (for datafiles, tablespaces, control files, and online redo logs).

Templates can be used just like scripts, but they are more powerful than scripts because you have the option of cloning a database. Cloning saves time by copying a seed database's files to the correct locations.

Templates are stored in the following directory:

```
ORACLE_HOME/assistants/dbca/templates
```

Advantages of Using Templates

Using templates has the following advantages:

- Time saving. If you use a template you do not have to define the database.
- Easy Duplication. By creating a template containing your database settings, you can easily create a duplicate database without specifying parameters twice.
- Easy editing. You can quickly change database options from the template settings.
- Easy sharing. Templates can be copied from one machine to another.

Types of Templates

Templates are divided into the following types:

- Seed templates
- Non-seed templates

The characteristics of each are shown in Table 2–1.

Table 2–1 DBCA Template Types

Type	File Extension	Includes Datafiles	Database Structure
Seed	`.dbc`	Yes	This type of template contains both the structure and the physical datafiles of an existing (seed) database. Your database starts as a copy of the seed database, and requires only the following changes: - Name of the database - Destination of the datafiles - Number of control files - Number of redo log groups - Initialization parameters Other changes can be made after database creation using custom scripts that can be invoked by DBCA, command-line SQL statements, or the Oracle Enterprise Manager. The datafiles and online redo logs for the seed database are stored in a compressed format in a file with a `.dfj` extension. The corresponding `.dfj` file's location is stored in the `.dbc` file.
Non-seed	`.dbt`	No	This type of template is used to create a new database from scratch. It contains the characteristics of the database to be created. Non-seed templates are more flexible than their seed counterparts because all datafiles and online redo logs are created to your specification, and names, sizes, and other attributes can be changed as required.

DBCA Templates Provided by Oracle

Oracle provides templates for the environments shown in Table 2–2.

Table 2–2 DBCA Template Environments

Environment	Description of Environment
Data Warehouse	Users perform numerous, complex queries that process large volumes of data. Response time, accuracy, and availability are key issues. These queries (`SELECT` statements) range from a fetch of a few records to queries that sort thousands of records from many different tables.
Transaction Processing	Many concurrent users perform numerous transactions that require rapid access to data. Availability, speed, concurrence, and recoverability are key issues. Transactions consist of reading (`SELECT` statements), writing (`INSERT` and `UPDATE` statements), and deleting (`DELETE` statements) data in database tables.
General Purpose	This template creates a database designed for general use. It combines features of both the DSS and OLTP database templates.
Custom Database	This template allows you maximum flexibility in defining a database.

Creating Templates Using DBCA

The Template Management window provides you with the option of creating or deleting a template. The DBCA saves templates as XML files.

To create a database template, select one of the following options:

- **From an existing template**

Using an existing template, you can create a new template based on the pre-defined template settings. You can add or change any template settings such as initialization parameters, storage parameters, or whether to use custom scripts.

- **From an existing database (structure only)**

  You can create a new template that contains structural information from an existing database, including database options, tablespaces, datafiles, and initialization parameters. User-defined schema and their data *will not* be part of the created template. The source database can be either local or remote. Choose this option when you want the new database to be structurally similar to the source database, but not contain the same data.

- **From an existing database (structure and data)**

  You can create a new template that has both the structural information and physical datafiles of an existing database. Databases created using such a template are identical to the source database. User-defined schema and their data *will* be part of the created template. The source database must be local. Choose the option when you want to create an exact replica of the source database.

When creating templates from existing databases, you can choose to translate file paths into Optimal Flexible Architecture (OFA) or maintain existing file paths. Using OFA is recommended if the machine on which you plan to create the database has a different directory structure. Standard file paths can be used if the target machine has a similar directory structure.

Deleting DBCA Templates

The Template Management window enables you to view and delete existing templates. When you delete a template, it is no longer available for creating a new database or a new template. Select a template and click Delete to remove it from the list of current templates.

Configuring Automatic Storage Management with DBCA

In the Operations window, select **Configure Automatic Storage Management** to configure Automatic Storage Management (ASM). For a brief overview of ASM, see "Advanced Installation" on page 2-3. For more detailed information, see Appendix A, "Automatic Storage Management" and *Oracle Database Administrator's Guide*.

Upgrading a Database

To upgrade an existing database to the current release of the Oracle Database, use the Database Upgrade Assistant (DBUA).

Overview of the Database Upgrade Assistant

The Database Upgrade Assistant (DBUA) guides you through the upgrade process and configures the database for the new release. The DBUA automates the upgrade process and makes appropriate recommendations for configuration options such as tablespaces and redo logs.

The DBUA provides support for Real Application Clusters (RAC) and Automatic Storage Management (ASM).

Pre-Upgrade Checks

The Database Upgrade Assistant does not begin the upgrade until it completes all of the following pre-upgrade steps:

- Checks for any invalid user accounts or roles
- Checks for any invalid datatypes or invalid objects
- Checks for any desupported character sets
- Checks for adequate resources, including rollback segments, tablespaces, and free disk space
- Checks for any missing SQL scripts needed for the upgrade
- Backs up all necessary files (optional).

Automated Upgrade Tasks

After completing the pre-upgrade steps, the Database Upgrade Assistant automatically performs the following tasks:

- Modifies or creates new required tablespaces
- Invokes the appropriate upgrade scripts
- Archives redo logs
- Disables archiving during the upgrade phase

While the upgrade is running, the DBUA shows the upgrade progress for each component. The DBUA writes detailed trace and log files and produces a complete HTML report for later reference. To enhance security, the DBUA automatically locks new user accounts in the upgraded database. The DBUA then proceeds to create new configuration files (initialization parameter and listener files) in the new Oracle home.

Support for Real Application Clusters

The Database Upgrade Assistant is fully compliant with the Real Application Clusters (RAC) environment. In a RAC environment, the DBUA upgrades all the database and configuration files on all nodes in the cluster.

> **Note:** On Windows operating systems, the DBUA does not support a direct upgrade of Oracle Parallel Server version 8.1.7 databases to Oracle Database 10*g* with RAC. You must first manually upgrade the Oracle Parallel Server database to Oracle Real Application Clusters Oracle9*i* release 2 (9.2), and then upgrade it to Oracle Database 10*g* with Real Application Clusters (RAC) using the DBUA.

Support for Automatic Storage Management

The DBUA supports upgrades of databases that use Automatic Storage Management (ASM). If an ASM instance is detected, you have the choice of updating both the database and ASM or only the ASM instance.

Support for Silent Mode

The Database Upgrade Assistant supports a silent mode of operation in which no user interface is presented to the user. Silent mode enables you to use a single statement for the upgrade.

Oracle Database Version Restrictions

The Database Upgrade Assistant supports the following versions of the Oracle Database for upgrading to Oracle Database 10*g*:

- Oracle Database version 8.0.6 (single instance databases only)
- Oracle8*i* release 8.1.7 (single instance and Real Application Clusters databases)
- Oracle9*i* release 1 (9.0.1) (single instance and Real Application Clusters databases)
- Oracle9*i* release 2 (9.2.0) (single instance and Real Application Cluster databases)

If your database version is not in this list, you need to upgrade first to the closest release listed. You can then upgrade the database to the current release.

Starting the Database Upgrade Assistant

If you install Oracle Database software only and specify that you are upgrading an existing database to the new Oracle release, then the Database Upgrade Assistant (DBUA) is launched automatically at the end of the software install. You can then continue as described in "Upgrading the Database Using the Database Upgrade Assistant" on page 2-17.

If you perform a software-only install and do not upgrade your database at that time, then you can do so later by launching the DBUA.

Be aware of the following before you begin using the DBUA:

- You must run the Oracle Net Configuration Assistant before running the DBUA.
- If the database instance is not running, the DBUA will try to start the instance with the default initialization parameter file. If that fails, you will be prompted to provide the name of the correct initialization parameter file or to start the instance. If the instance is already up and running, the DBUA connects to it.

Note: If you stop the upgrade, but do not restore the database, then you should not restart the DBUA until you start up the existing database in `UPGRADE` mode using the 10.2 server. You cannot go back to the original server unless you restore your database.

If you restore your database manually (not using the DBUA), then remove the following file before starting the DBUA:

```
$10.2OracleHome/cfgtoollogs/dbua/logs/Welcome_<SID>.txt
```

The presence of this file indicates to the DBUA that this is a re-run operation.

To launch the DBUA on the Windows operating system choose the following menu option:

Start, Programs, Oracle - *HOME_NAME*, Configuration and Migration Tools, Database Upgrade Assistant

To launch the DBUA on a UNIX or Linux operating system, enter the following at a command prompt:

```
dbua
```

This command is typically found in your `$ORACLE_HOME/bin` directory.

See Also: *Oracle Database Upgrade Guide* for more information about the Database Upgrade Assistant

When the DBUA starts, the Welcome screen appears. At the Welcome screen, you may begin the upgrade process.

Upgrading the Database Using the Database Upgrade Assistant

Complete the following steps to upgrade a database using the Database Upgrade Assistant (DBUA). If you need help at any screen or want to consult more documentation about the DBUA, then click the **Help** button to open the online help.

1. At the Welcome screen of the DBUA make sure the database being upgraded meets the specified conditions. Then, click **Next**.
2. If an ASM instance is detected on the system, then the Upgrade Operations page is displayed. From this page you can choose to upgrade only the ASM instance or the database. If you choose to upgrade the database and if the database is using ASM, then the DBUA will prompt you whether to upgrade the ASM instance along with the database. Oracle recommends that you upgrade the database and ASM in separate DBUA sessions, in separate Oracle homes.
3. At the Selecting a Database Instance screen, select the database you want to upgrade from the Available Databases table. Then, click **Next**.

   You can select only one database at a time. If you are running the DBUA from a user account that does not have `SYSDBA` privileges, then enter the user name and password credentials to enable `SYSDBA` privileges for the selected database.

   The DBUA analyzes the database, performing pre-upgrade checks and displaying warnings as necessary:

   - It checks for any online redo log files of a size less than 4 MB. If such files are found, then the DBUA gives the option to drop or create new online redo log files.
   - It checks the initialization parameter file for any obsolete or deprecated initialization parameters.
4. At the Creating the SYSAUX Tablespace screen, specify the attributes for the `SYSAUX` tablespace, which is added automatically to all new Oracle Database 10*g* databases you create. Then, click **Next**.

   See Also: *Oracle Database Administrator's Guide* for more information about the `SYSAUX` tablespace

   Many of the attributes of the `SYSAUX` tablespace are set automatically and cannot be modified. For example, the `SYSAUX` tablespace is set to use Automatic Segment-Space Management. Nevertheless, you can specify the location of the data file, the default size of the `SYSAUX` tablespace, and its autoextend attributes.

   Note: If you specify an existing datafile for the `SYSAUX` tablespace, then you must select **Reuse Existing File Name**. Otherwise, the DBUA alerts you that the file already exists.
5. When you upgrade your database to the new Oracle Database 10*g* release, many of the PL/SQL modules in the database become invalid. By default, the Oracle Database will recompile invalid PL/SQL modules as they are used, but this takes

time and can result in poor performance. To eliminate these performance issues, select **Recompile invalid objects at the end of upgrade** from the Recompile Invalid Objects screen. All the invalid PL/SQL modules will be recompiled immediately after the upgrade is performed.

If you have multiple CPUs, the DBUA automatically adds a **Degree of Parallelism** menu to the Recompile Invalid Objects screen. Parallel processing reduces the time it takes to recompile all the invalid PL/SQL modules in your database. The DBUA automatically sets the degree of parallelism to one less than the number of available CPUs. You can select a different value from the menu.

Note: Selecting **Recompile invalid objects at the end of upgrade** is equivalent to running the `ORACLE_HOME/rdbms/admin/utlrp.sql` script, which is used to recompile stored PL/SQL and Java code.

If the database is in `ARCHIVELOG` mode, you can change it to `NOARCHIVELOG` mode.

Then, click **Next**.

6. At the Choosing a Database Backup Procedure screen, specify whether or not you want the DBUA to back up your database for you. Then, click **Next**.

   If you choose not to use the DBUA for your backup, then it is assumed that you have already backed up your database using your own backup procedures. It is strongly recommended that you back up your database before the upgrade. If errors occur during the upgrade, then you may need to restore the database from the backup.

   Note: The DBUA does not back up ASM databases. You must manually back them up on your own.

   If you use the DBUA to back up your database, then the DBUA makes a copy of all database files in the directory that you specify in the **Backup Directory** field. The DBUA performs this consistent backup automatically after it shuts down the database and before it begins the upgrade process. The backup does not compress the database files; the backup directory must be a valid file system path. You cannot specify a raw device for the cold backup files.

   In addition, the DBUA creates a batch file in the specified directory. Use the batch file appropriate to your system to restore the database files:

   - Windows: `db_name_restore.bat`
   - UNIX: `db_name_restore.sh`

7. At the Management Options screen, you have the option of setting up your database so it can be managed with Enterprise Manager.

   Note: The Management Options screen described in this step is not displayed if an Oracle Database 10*g* release 10.1 database is already being monitored with Enterprise Manager.

a. The DBUA checks to see if the Oracle Management Agent has been installed on the host computer. If it has, select the **Grid Control** option and select an Oracle Management Service from the list. When you finish installing the Oracle Database, the database is automatically available as a managed target within the Oracle Enterprise Manager Grid Control.

b. To configure the database so it can be managed with the Oracle Enterprise Manager Database Control, select the **Database Control** option. This provides Web-based features for monitoring and administering the single-instance or cluster database you are installing.

c. Select **Enable Email Notifications** to enable the `SYSMAN` user (the default Super Administrator and owner of the Management Repository schema) to receive e-mail notification when a metric for a specified condition reaches a critical or warning threshold.

d. Select **Enable Daily Backups** to use the Oracle-suggested backup strategy to back up your entire database with a minimum amount of configuration. At the start time you enter, Enterprise Manager backs up the database to the Flash Recovery Area that you specify later on the Recovery Configuration screen of the DBUA.

e. Click **Next**.

8. At the Database Credentials screen, secure your database with passwords for your Enterprise Manager accounts. You can set a single password for each of the listed Enterprise Manager user accounts, or you can enhance the security of the accounts by providing unique passwords for each user.

9. At the Recovery Configuration screen, specify a Flash Recovery Area and enable archiving. It is important to configure the database so you can recover your data in the event of a system failure.

   The Flash Recovery Area can be used to recover data that would otherwise be lost during a system failure; this location is also used by Enterprise Manager if you have enabled local management and daily backups on the Management Options screen in the DBUA.

10. At the Network Configuration for the Database screen, there are two tabs:

    - The **Listeners** tab is displayed if you have more than one listener in the Oracle Database 10*g* release 10.2 Oracle home. Select the listeners for which you would like to register the upgraded database.

    - The **Directory Service** tab is displayed if you have directory services configured in the Oracle Database 10*g* release 10.2 Oracle home. You can select whether or not to register the upgraded database with the directory service.

11. At the Summary screen, make sure all of the specifications are correct. The DBUA lists the initialization parameters that will be set for the database during the upgrade. The `COMPATIBLE` initialization parameter will be set to at least `9.2.0`. Click **Back** to correct any errors, then click **Finish**.

12. A Progress dialog appears and the DBUA begins to perform the upgrade.

    You may encounter error messages with **Ignore** and **Abort** choices:

    - **Ignore** - Ignores the error and proceeds with the upgrade. The ignored errors are logged and shown later in the summary. After the upgrade is complete, you can fix the problem, restart the DBUA, and complete the skipped steps.

    - **Abort** - Stops the upgrade process.The DBUA prompts you to restore the database if the database backup was taken by the DBUA. After the database has been restored, correct the error and restart the DBUA to perform the upgrade again. If you do not restore the database, then the DBUA leaves the database in its current state so that you can proceed with a manual upgrade.

After the upgrade has completed, the following message is displayed:

```
Upgrade has been completed. Click the "OK" button to see the results of the
upgrade.
```

Click **OK**. The Checking Upgrade Results screen appears.

13. At the Checking Upgrade Results screen, examine the results of the upgrade, manage the passwords in the upgraded database, and, if necessary, restore the original database settings.

The upgrade results summary describes the original and upgraded databases and changes made to the initialization parameters. The upgrade results also include a Step Execution Summary that describes the steps performed during the database upgrade. The summary provides each step name, the log file for the step, and the status. In some cases, you can click the status to display details about the execution step. The Step Execution Summary also includes the directory where the various log files are stored after the upgrade. You can examine any of these log files to obtain more details about the upgrade process.

Note: An HTML version of the Upgrade Results is also saved in the log files directory.

The Password Management section of the screen enables you to unlock and set passwords for various users in the newly upgraded database. Click **Configure Database Passwords** to display the Password Management dialog box. The Password Management dialog box enables you to change the default password for a user after you upgrade the database. For security reasons, normally only the following users are unlocked:

- SYS
- SYSTEM

If you have enabled Local Management with Enterprise Manager, then the SYSMAN and DBSNMP accounts are also unlocked. These accounts allow Enterprise Manager to gather monitoring data from the database.

If you have enabled Central Management with Enterprise Manager, then the DBSNMP account is unlocked, as well as the SYS and SYSTEM user accounts.

Note: To prevent unauthorized use of the database, change all user passwords immediately after you upgrade your database.

If you are not satisfied with the upgrade results, click **Restore**. Depending on the method you used to back up your database, the restore operation performs one of the following tasks:

- If you used the DBUA to back up your database, then clicking **Restore** restores the original database and the original database settings from the backup.
- If you used your own backup procedure to back up the database, then clicking **Restore** restores only the original database settings. To restore the database itself, restore the backup you created with your own backup utilities.

If you are satisfied with the upgrade results, click **Exit** to quit the DBUA. The DBUA removes the entry of the upgraded database from the old `listener.ora` file and reloads the listener of the old database.

14. At the **Changes in Default Behavior** screen, the DBUA displays changes in behavior of Oracle Database 10*g* from that of previous releases. In some cases the default values of some initialization parameters have changed. In other cases some new behavior or requirement has been introduced that may affect current scripts or applications.
15. See the *Oracle Database Upgrade Guide* for information on additional tasks that should be completed after upgrading a database.

WARNING: If you retain the old Oracle software, never start the upgraded database with it. Only start the database with the executables in the new Oracle Database installation. Also, before you remove the old Oracle environment, make sure you relocate any datafiles in that environment to the new Oracle Database environment. See the *Oracle Database Administrator's Guide* for information about relocating datafiles.

Installation: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter and includes annotated screen shots.

To view the Installation OBE, point your browser to the following location:

`http://www.oracle.com/technology/obe/10gr2_2day_dba/install/install.htm`

3

Getting Started with Oracle Enterprise Manager

This chapter introduces you to the Oracle Enterprise Manager Database Control, the Web-based interface for managing an Oracle database. Understanding this tool is essential to managing the Oracle database as described in the rest of this book.

This chapter contains the following topics:

- Introduction to the Oracle Enterprise Manager Database Control
- Starting and Stopping the Oracle Enterprise Manager Console
- Accessing the Oracle Enterprise Manager Database Control
- Granting Database Control Administrative Privileges
- Setting Up Enterprise Manager Preferences
- SQL Statements and Other Management Tools: SQL*Plus and iSQL*Plus
- Managing Your Database: A Roadmap
- Getting Started: Oracle by Example Series

Introduction to the Oracle Enterprise Manager Database Control

The Web-based Database Control serves as the primary tool for managing your Oracle database and sets a new standard in ease-of-use. It is installed with the database.

The following topics introduce you to the Oracle Enterprise Manager Database Control:

- Database Management Features
- Online Help
- Navigational Features

Database Management Features

From the Oracle Enterprise Manager Database Control, you can perform administrative tasks such as creating schema objects (tablespaces, tables, and indexes), managing user security, backing up and recovering your database, and importing and exporting data. You can also view performance and status information about your database.

Online Help

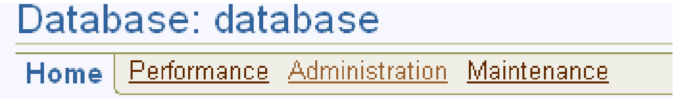
You can access context sensitive online Help by clicking **Help** displayed on every page. On any help page, click **View Contents** for links to all help topics. A search facility enables you to search the contents of Help.

Navigational Features

Navigational features of Oracle Enterprise Manager include the following:

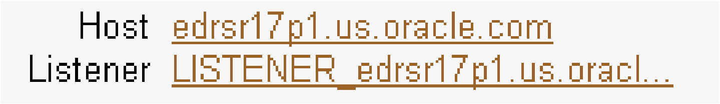
- Property pages on each home page help you organize management tasks into distinct categories; for example, click Maintenance to perform tasks associated with backing up and maintaining your data.

Figure 3–1 Database Home Page Navigational Property Pages



- Drill down links that provide increasing levels of detail.

Figure 3–2 Database Home Page Drill Down Links

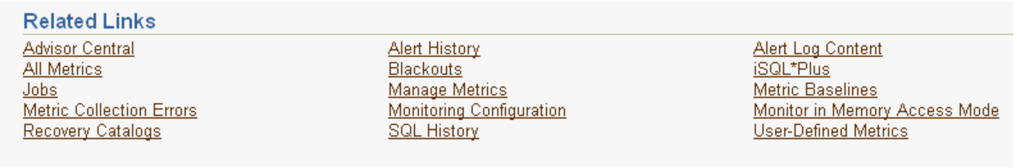


Host edrsr17p1.us.oracle.com
Listener LISTENER_edrsr17p1.us.oracl...

Clicking a link shown in Figure 3–2 provides additional levels of detail about the database host name or listener.

- Suggested related links for more information.

Figure 3–3 Database Home Page Related Links



Starting and Stopping the Oracle Enterprise Manager Console

To access the Oracle Enterprise Manager Console from a client browser, the `dbconsole` process needs to be running on the server. The dbconsole process is automatically started after installation.

However, in the event of a system restart, you can start it manually at the command line or start it as a service in Windows.

To start the `dbconsole` process from the command line:

1. Navigate into your `ORACLE_HOME/bin` directory.
2. Run the following statement:

```
    ./emctl start dbconsole
```

Additionally, you can stop the process and view its status.

To stop the `dbconsole` process:

```
./emctl stop dbconsole
```

To view the status of the `dbconsole` process:

```
./emctl status dbconsole
```

Starting dbconsole in Windows

In Windows, in addition to using the command line, you can start the `dbconsole` process as a service.

To start `dbconsole` as a service:

1. From the main menu, click Start, Control Panel, Administrative Tools, Services. The Services page appears.
2. Oracle services begin with `Oracle`. The `dbconsole` service is listed as OracleDBConsole*`ORACLE_SID`*, where *`ORACLE_SID`* is your SID. The status of this process is listed in the Status column, either Started or Stopped.

   Double click the service. The property page appears.
3. In the properties page, ensure that the Startup Type is either Manual or Automatic and not Disabled. Click **Start**, if the process is not already started. Click **OK**.

You can also use the Services page to stop the process.

Accessing the Oracle Enterprise Manager Database Control

At the end of a successful database creation, Oracle Enterprise Manager is automatically launched in your Web browser and the database is started. To access Enterprise Manager Database Control at other times, make sure the dbconsole process is running on the server, then follow these steps:

1. Point your Web browser to the following URL:

   ```
   http://hostname:portnumber/em
   ```

   For example, if you installed the database on a host computer named `comp42`, and the installer indicated that your Enterprise Manager Console HTTP port number is 5500 (also recorded in the `$ORACLE_HOME/install/portlist.ini` file), enter the following URL:

   ```
   http://comp42:5500/em
   ```

   If the database is up, Enterprise Manager displays the Database Control Login page.

   If the database is down and needs to be re-started, Enterprise Manager displays the Startup/Shutdown and Perform Recovery page. If this is the case, click Startup/Shutdown and enter the host and target database login usernames and passwords. For the database user and password, use `SYS` and the password you specified during installation.
2. Click **OK** to start the database. In the Confirmation screen, click **YES** to start the database in open mode.

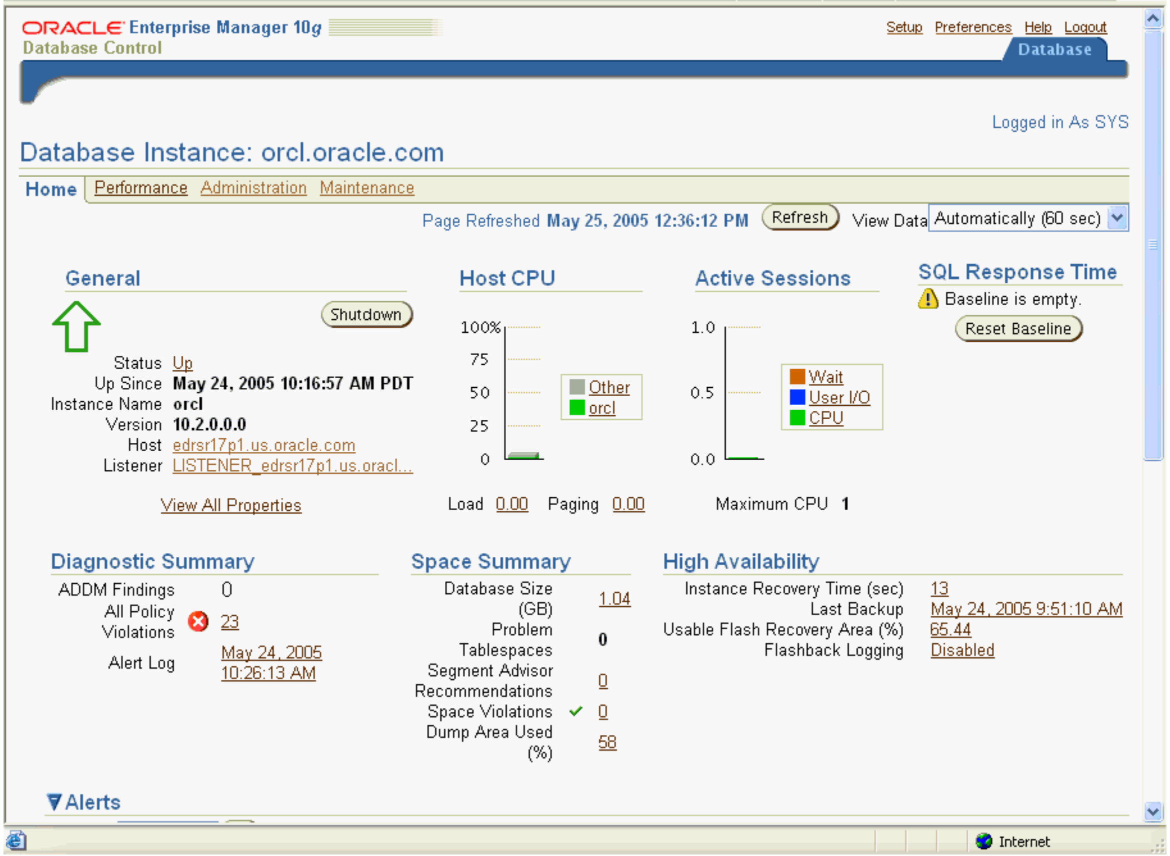
3. Log in to the database using a username that is authorized to access the Database Control. This initially could be `SYS` or `SYSTEM`. Use the password you specified for the account during the database installation.

   Enterprise Manager displays the Database Home page (Figure 3–4).

   The property pages across the top of the page enable you to access performance, administration, and maintenance pages for managing your database. The functionality provided by these pages is discussed in other chapters of this book.

   The various sections of the Database Home page and related links provide information about the database's environment and health. For example, the **Alerts**, **Related Alerts**, and **Diagnostic Summary** sections warn you of errors and performance problems that are impacting the operation of your database. You can click the provided links to see more detail about the problem area, and even to obtain recommendations for resolving the problem. This is discussed in Chapter 10, "Monitoring and Tuning the Database".

Figure 3–4 Database Home Page



Logged in As SYS

Database Instance: orcl.oracle.com

Page Refreshed May 25, 2005 12:36:12 PM

View Data Automatically (60 sec)

General

Status Up
Up Since May 24, 2005 10:16:57 AM PDT
Instance Name orcl
Version 10.2.0.0.0
Host edrsr17p1.us.oracle.com
Listener LISTENER_edrsr17p1.us.oracl...

Host CPU

Load 0.00 Paging 0.00

Active Sessions

Maximum CPU 1

SQL Response Time

⚠ Baseline is empty.

Diagnostic Summary

ADDM Findings	0
All Policy Violations	☒ 23
Alert Log	May 24, 2005 10:26:13 AM

Space Summary

Database Size (GB)	1.04
Problem Tablespaces	0
Segment Advisor Recommendations	0
Space Violations	✓ 0
Dump Area Used (%)	58

High Availability

Instance Recovery Time (sec)	13
Last Backup	May 24, 2005 9:51:10 AM
Usable Flash Recovery Area (%)	65.44
Flashback Logging	Disabled

Granting Database Control Administrative Privileges

When you log in to the Oracle Enterprise Manager Database Control using the `SYS` or `SYSTEM` user account, you are logging in as the Oracle Enterprise Manager super user. These are the only accounts that are automatically granted the roles and privileges required to access all the management functionality provided the Database Control.

To grant management access to other database users, use the following procedure:

1. Start your Web browser and log in to the Database Control as the `SYS` or `SYSTEM` database user.
2. Click **Setup** at the top of the Database Home page.
3. Click **Administrators** in the left navigation bar.

   If you need to create a new administrator, you must first create the user and then assign administrative privileges as described in this section. To create a new user, see "Administering Database Users" on page 7-8.
4. Click **Create** to create a new Enterprise Manager user by assigning the management privileges to an existing database user.
5. Click the flashlight icon next to the Name field and select an existing database user.
6. Enter the password for the selected user and click **Finish**.

Enterprise Manager assigns the management privileges to the selected user. The selected database user is now included in the list of management users on the Setup Administrators page. All the users shown on this page can log in to the Database Control and perform database management tasks.

Setting Up Enterprise Manager Preferences

Enterprise Manager enables you to set up preferences that help you manage the database. These include the following:

Notification

These settings enable Oracle to e-mail you alerts. Alerts are notifications of when a database is in an undesirable state and needs your attention. By default, the Enterprise Manager home page lists all alerts. However, set up is required for e-mail notification. For more information about alerts and setting up notifications, see "Setting up Direct Alert Notification" in Chapter 10, "Monitoring and Tuning the Database".

Blackout Administration

When you have planned downtime for database maintenance, you can indicate that you do not want false alerts to be sent to you. To do so, you define a blackout period. See "Defining Blackout Periods" on page 3-5.

Preferred Credentials

Enterprise Manager can automatically execute many routine administrative tasks, such as backups. This is done using sophisticated job scheduling system built into the Enterprise. To keep your environment secure, setting up tasks for automatic execution in Enterprise Manger requires you to provide login information for the machine and database. To avoid having to enter this information every time you create a job or task, Enterprise Manager enables you to save this information as preferred credentials. Preferred credentials are stored in the database in encrypted mode to protect them from unauthorized use. See "Setting Preferred Credentials" on page 3-6.

Defining Blackout Periods

When you plan to bring your database down for maintenance, you can indicate that you do not want alert notifications to be sent to you. Alerts are notifications of when the database is in an undesirable state and needs your attention. For more information about alerts, see "Alerts" on page 10-1.

To define a blackout time period, follow these steps

1. From the home page, click **Setup** at the top of the page. The Setup page appears.
2. Click **Blackouts** in the left hand pane. The Blackouts page appears.
3. Click **Create** to start the Create Blackout wizard. On the Properties page, name your blackout, enter a reason, and select the target that has planned downtime. Click **Next**.
4. On the Schedule page, enter the start time of your planned blackout, or choose Immediately if you are bringing the database down now. Also select the duration of the blackout, either as indefinite, as a length or time, or until a time in the future. Under Repeating, accept the default of Do Not Repeat. If you want to repeat the blackout periodically, you can select a repeat frequency in the pull down menu. Click **Next**.
5. The review page appears. Review what you have entered, and click **Finish**. You can click Back to change a setting.

Setting Preferred Credentials

You can have Enterprise Manager automatically fill in host and database login credentials for you, such as when you schedule jobs and tasks to perform administrative operations like backup and recovery. For security, Oracle stores preferred credentials in encrypted mode.

To set preferred credentials for the database, do the following:

1. From the home page, click **Preferences** at the top of the page. The Preferences page appears.
2. Click **Preferred Credentials** in the left hand pane. The Preferred Credentials page appears. For the Database target, click the icon under Set Credentials. The Database Preferred Credentials page appears.
3. For your database, enter the credentials for Normal Username/Password, SYSDBA Username/Password, and Host Username/Password.
4. Click **Test** to test your credentials. You should get a confirmation message if your credentials can be verified.
5. Click **Apply** to apply the changes.

SQL Statements and Other Management Tools: SQL*Plus and *i*SQL*Plus

The underlying operations in Enterprise Manager are performed using Structured Query Language (SQL) statements. SQL is an English-like computer programming language for accessing and manipulating data contained in a database. It is an industry-standard language for accessing databases.

The following is an example of a SQL query:

```
SELECT COUNTRY_ID, COUNTRY_NAME FROM HR.COUNTRIES;
```

SQL is a powerful language and can be used to perform a variety of database administrative tasks. While performing these tasks, you can click Show SQL to see the underlying SQL statements being issued.

For more information on SQL and on using SQL for database administration, see *Oracle Database SQL Reference*.

In addition to Enterprise Manager, you can use other Oracle tools to issue SQL statements such as SQL*Plus and its Web version *i*SQL*Plus. These tools enable you to perform the same database management operations, as well as to query, insert, update or delete data directly in the database.

Starting SQL*Plus

SQL*Plus is a command line program you use to issue SQL statements to an Oracle database. You can issue these statements or in batch. SQL*Plus is installed with the database and is located in your `$ORACLE_HOME/bin` directory.

To start SQL*Plus, enter the following at the command line on all platforms:

```
sqlplus username/password
```

You can use the `SYS` administrative user and password you set up during installation. SQL*Plus starts and connects you to the default database. At the SQL prompt, you can enter statements that perform administrative tasks such as database startup and shutdown. You can also query, insert, update, and delete data.

On Windows, you can use the SQL*Plus Windows GUI to administer the database. From the main menu, select Start, Programs, Oracle-OraDB10g_home1, Application Development, SQL*Plus.

For more information about SQL*Plus, see *SQL*Plus User's Guide and Reference*.

For more information about SQL statements, see *Oracle Database SQL Reference*.

Starting *i*SQL*Plus

*i*SQL*Plus is the Web version of SQL*Plus. Before you can connect to it with a URL, you must start the *i*SQL*Plus application server. To do so, enter the following at the command line:

```
isqlplusctl start
```

After the server is started, you can connect to *i*SQL*Plus with a URL such as

```
http://machine_name:5560/isqlplus
```

Alternatively, you can connect to *i*SQL*Plus using Enterprise Manger. To do so from the Enterprise Manager home page, click **iSQL*Plus** under Related Links.

For more information about *i*SQL*Plus, see *SQL*Plus User's Guide and Reference*.

Managing Your Database: A Roadmap

The following are some specific tasks to perform to become familiar with your database. They are presented as links to the chapters that contain information about these tasks and other related database management tasks.

1. Start up your instance and open the database. At the end of installation, your instance is already started and your database is open. In the future there will be times, perhaps for doing database maintenance or because of a power or media failure, that you shut down your database or instance and later restart it. See "Shutting Down and Restarting the Instance and Database with Enterprise Manager" on page 5-6.
2. Optionally configure the network environment to enable clients to connect to your database. See Chapter 4, "Configuring the Network Environment".

3. Review your database storage structure: tablespaces and datafiles, redo log files, and control files. See Chapter 6, "Managing Database Storage Structures".
4. Review, unlock, and reset passwords as necessary for initial database users. Create new users and assign privileges and roles to them. See Chapter 7, "Administering Users and Security".
5. Create users and necessary schema objects, including tablespaces and tables. Populate tables with data. See Chapter 8, "Managing Schema Objects".
6. Review a backup strategy for the database and back up the database. See Chapter 9, "Performing Backup and Recovery".
7. Enable archiving of redo logs, if not already done. See "Configuring Your Database for Basic Backup and Recovery" on page 9-4.
8. Monitor database performance, diagnose performance problems and tune the database as necessary. See Chapter 10, "Monitoring and Tuning the Database".
9. Keep Oracle software up-to-date with the latest patch releases. See Chapter 11, "Managing Oracle Software".

Getting Started: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter, and includes annotated screen shots.

To view the Getting Started OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_
dba/gettingstarted/gettingstarted.htm
```

4

Configuring the Network Environment

After installing your Oracle software and database, you have a fully functional database with a client/server network environment that has been minimally configured.

This chapter contains the following topics to help you completely configure your client/server network:

- Understanding Network Configuration
- Tools for Network Configuration
- Viewing Listener Configuration
- Starting and Shutting Down the Listener
- Configuring Oracle Networking on Client Machines
- Networking: Oracle by Example Series

Understanding Network Configuration

A client is any application that connects to the Oracle database to send or retrieve data. An Oracle client application can reside on any machine provided it has Oracle client software installed.

Oracle Net is a software component that resides on the client and on the Oracle database server. It establishes and maintains the connection between the client application and the server, and exchanges messages between them using industry standard protocols.

For the client application and a database to communicate, the client application must specify location details for the database it wants to connect to, and the database must provide some sort of identification or address.

> **See Also:** *Oracle Database Net Services Administrator's Guide* for more guidance in understanding Net Services and for more configuration details.

Oracle Net Listener Configuration

On the database server, the Oracle Net listener, commonly known as the listener, is a process that listens for client connection requests. The default listener configuration file is called `listener.ora`, and it is located in the *`ORACLE_HOME`*`/network/admin` directory. The file contains a protocol address that identifies the database. This address defines the protocol the listener is listen on and any other protocol specific information. For example, the listener could be configured to listen at the following protocol address:

```
(DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=my-server) (PORT=1521)))
```

This example shows a TCP/IP protocol address that specifies the host machine of the listener and a port number.

The `listener.ora` file is automatically configured during installation, but can be configured with Enterprise Manager. For more information, see "Viewing Listener Configuration" and "Starting and Shutting Down the Listener" on page 4-4.

Client Configuration

When a client requests a connection, the listener on the server brokers the request and forwards it to the Oracle database.

The client uses a connect descriptor to specify the database it wants to connect to. This connect descriptor contains a protocol and a database service name. A database can have multiple services defined, so a specific service must be specified for the connection. In a preconfigured database, there is only one service, the name of which defaults to the global database name.

The following example shows a connect descriptor that enables clients to connect to a database service called `mydb.us.acme.com`:

```
(DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp)(HOST=my-server) PORT=1521))
   (CONNECT_DATA=
      (SERVICE_NAME=mydb.us.acme.com)))
```

You can define your connect descriptors in your `tnsnames.ora` file on the client machine, located in the *`ORACLE_HOME`*`/network/admin` directory. You can use Oracle Net Manager to configure this file as described in "Configuring Oracle Networking on Client Machines" on page 4-5.

Connection Requests

Users initiate a connection request by providing a connect string. A connect string includes a username and password, and a connect identifier. This connect identifier can be the connect descriptor itself, or a name that resolves to the connect descriptor. One of the most common connect identifiers is a net service name, a simple name for a service. The following example shows a connect string that uses the net service name `mydb` as the connect identifier.

```
CONNECT hr/hr@mydb
```

While the connect identifier shown here is relatively simple, others can be long and inconvenient to use if your environment configuration is more complex. With long connect identifiers, you can map the net service name to the connect descriptor. This mapping information is stored in one or more repositories of information that are accessed with naming methods.

Naming Methods

Oracle Net provides support for the following naming methods:

- Local Naming

- Directory Naming
- Easy Connect Naming
- External Naming

The preconfigured database uses local naming. Only this form of naming is used in this book.

Local Naming

The local naming method stores connect descriptors, identified by their net service name, in a configuration file on the client named `tnsnames.ora`. This file is located in the *`ORACLE_HOME`*`/network/admin` directory.

Directory Naming

The directory naming method stores connect identifiers in a centralized, LDAP-compliant directory server.

Easy Connect Naming

The easy connect naming method enables clients to connect to an Oracle database server by using only a TCP/IP connect string consisting of a host name and optional port and service name:

```
CONNECT username/password@host[:port][/service_name]
```

For example:

```
CONNECT hr/hr@my-server:1521/mydb
```

The easy connect naming method requires no configuration.

External Naming

The external naming method stores net service names in a supported non-Oracle naming service. These supported third-party services include:

- Network Information Service (NIS) External Naming
- Distributed Computing Environment (DCE) Cell Directory Services (CDS)

Tools for Network Configuration

Oracle enables you to manage your network configuration with the following tools: Oracle Net Configuration Assistant, Enterprise Manager, and Oracle Net Manager.

Oracle Net Configuration Assistant

The Oracle Universal Installer launches Oracle Net Configuration Assistant after the database is installed. Use Oracle Net Configuration Assistant to configure the listening protocol address and service information for an Oracle database.

During a typical database install, Oracle Net Configuration Assistant automatically configures a listener called `LISTENER` that has a TCP/IP listening protocol address for the database. If you do a custom install, Oracle Net Configuration Assistant prompts you to configure a listener name and protocol address of your choice.

Use Oracle Net Configuration Assistant for initial network configuration after database installation. Thereafter, you can use the Oracle Enterprise Manager and Oracle Net Manager to configure and administer your networks.

Enterprise Manager

Enterprise Manager enables you to manage your server-side network configuration with two pages: the Listener page and the Net Services Administration page.

The Listener page displays the listener status and enables you to shut it down. You can navigate to this page from the Home page by clicking the **Listener** link under General. See "Viewing Listener Configuration" in this chapter.

You can navigate to the Net Services page from the Home page by clicking **Listener** under General. The Listener page appears. Under Related Links, click **Net Services Administration**.

The Net Services Administration page enables you to do the following:

- Perform the following administrative tasks for a selected listener:
  - Show current status
  - Change status (start/stop listener)
  - Change tracing settings
  - Change logging settings
- Configure the following:
  - Listener
  - Local naming (`tnsnames.ora` files) on the machine running the database
  - Directory naming
- Search and sort local and directory naming entries
- Export directory naming entries to `tnsnames.ora` file

Oracle Net Manager

You can access Oracle Net Manager from the command line or for Windows platforms through the Start menu.

- For command line (UNIX, Linux, or Windows) run `netmgr`.
- On Windows click the **Start** button and select **Programs**, **Oracle - *oracle_home***, **Configuration and Migration Tools**, and then **Net Manager**

Oracle Net Manager provides the same features as Oracle Enterprise Manager with the addition of configuring profiles. To learn how to use this tool, see "Configuring Oracle Networking on Client Machines" on page 4-5.

Viewing Listener Configuration

The listener runs on your database server machine and brokers incoming client requests. With Enterprise Manager, you can view the status of the listener, which is set to start automatically whenever your server machine is restarted.

To view the status of the listener, navigate to the Enterprise Manager home page. Under General, click **Listener**. The Listener: *listener_name* page appears. Under General, note the listener status, which is either Up or Down.

This page also gives the listener version, its net address, and start time.

You can additionally view the listener status at the command line using the following:

```
lsnrctl status
```

Starting and Shutting Down the Listener

The Oracle listener is set up to start automatically whenever your server machine is restarted. However, when your system encounters unforeseen circumstances, or when you have manually stopped the listener, you can restart it at the command line. To do so, use the following:

```
lsnrctl start
```

You can use Enterprise Manager to stop the listener. To do so, navigate to the Listener: listener_name page by clicking **Listener** on the Home page. To shut down the listener, click **Stop**.

You can also stop it at the command line using the following:

```
lsnrctl stop
```

Configuring Oracle Networking on Client Machines

Client machines need to be configured before they can connect to an Oracle database. To configure the client machine, you must first install Oracle client software, which includes Oracle Net software.

This example shows you how to configure a Windows client with local naming, which involves adding a connect descriptor entry to the local `tnsnames.ora` file. The connect descriptor specifies the host name, protocol address, and service name of the database to connect to.

On Windows, you can use the Oracle Net Manager program to create a new net service entry in your `tsnames.ora` file as follows:

1. Start Oracle Net Manager by clicking the **Start** button and then selecting **Programs**, **Oracle**, **Configuration and Migration Tools**, and **Net Manager**.
2. From the Local tree, select **Service Naming** and click the plus (+) on the left hand side of the page.

   The Net Service Name Wizard appears.
3. Enter a net service name to identify it. The name you choose will appear under the Service Naming folder. Click **Next**.
4. Select the protocol to use to connect to your database. For example, you can accept the default of **TCP/IP (Internet Protocol)**. The database listener must be configured to use the same network protocol. Click **Next**.
5. Enter the host name of the database machine, such as `coconuts.island.com`, and the port number. Click **Next**.
6. Enter either the database service name or the database SID. In the **Connection Type** list, select either **Shared Server**, **Dedicated Server**, or **Database Default**. Shared connections require your database to be running in shared mode. If you choose Dedicated, then the database dedicates a single server process to serve your connection. If you are unsure or would like to use the default database connection type, select **Database Default.** Click **Next**.
7. The last screen of the wizard enables you to test the connection by using the information you entered. To do so, click **Test**. To log in to the database, you can use the `SYS` user and the password you specified when you configured your database. The wizard tells you if the connection test succeed or not.

8. Click **Finish**. The new service appears under the Service Naming folder. You can view or modify the configuration for the highlighted service in the Service Identification and Address Configuration sections.

Note that you can additionally run Oracle Net Manager from the command line using `netmgr` on Windows, Linux, and Unix platforms.

Networking: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter and includes annotated screen shots.

To view the Networking OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/network/network.htm
```

5 Managing the Oracle Instance

This chapter discusses the Oracle instance and database.

The following topics are covered:

- Overview of an Instance and Instance Management
- Shutting Down and Restarting the Instance and Database with Enterprise Manager
- Viewing and Modifying Initialization Parameters
- Managing Memory Parameters
- Instances: Oracle by Example Series

Overview of an Instance and Instance Management

The Oracle database is made of a set of operating system files containing data entered by users or applications and structural information about the database itself called metadata. Information is stored persistently in these files.

To enable users and applications to view or update data in the database, Oracle must start a set of processes, called background processes, and must allocate some memory to be used during database operation. The background processes and memory allocated by Oracle together make up an instance. An instance must be started to read and write information to the database. However, having a database is not necessary to run an instance.

When the database instance is not available, your data is safe in the database but it cannot be accessed by any user or application.

The properties of a database instance are specified using instance initialization parameters. When the instance is started, an initialization parameter file is read and the instance is configured accordingly.

This section presents some of the concepts of an instance and its management. The following topics are discussed:

- About Initialization Parameters
- Instance Memory Structure
- Oracle Background Processes
- Accessing the Database

About Initialization Parameters

Instance management involves configuring parameters that affect the basic operation of the database instance. These parameters are called initialization parameters. The Oracle database server reads these parameters at database startup and monitors them while the database is running. They are stored in memory, where many of them can be changed dynamically. There are two types of parameter files, and whether these dynamic changes are persistent across database shutdown and startup depends upon the type of parameter file you are using.

Server Parameter File

The server parameter file is the preferred form of initialization parameter file, and is a binary file that can be written to and read by the database server. It *must not* be edited manually. It resides on the machine that Oracle is running on, and changes to it persist across shutdown and startup.

Text Initialization Parameter File

This type of initialization parameter file can be read by the database server, but it is not written to by the server. In this file, you can set initialization parameters with a text editor for them to be persistent across shutdown and startup.

Instance Memory Structure

The size of the instance memory structures affects the performance of the Oracle database server and is controlled by initialization parameters.

When a database is created with DBCA, the memory parameters are automatically set to optimal values based on the database workload, such as data warehouse, general purpose, or transaction processing. However, as your database usage expands, you can alter the settings of the memory parameters.

Oracle provides alerts and advisors to identify memory sizing problems and to help you determine appropriate values for memory parameters.

The System Global Area (SGA)

The SGA is a shared memory area that contains data and control information for the instance. Multiple users can share data within this memory area and avoid repeated, time-consuming access from physical disk. For optimal performance, the SGA should be large enough to avoid frequent disk reads and writes.

The SGA has several subcomponents as listed in:

Table 5–1 SGA Components

Component	Description
Buffer Cache	Before any data stored in the database can be queried or modified, it must be read from disk and stored in memory in the buffer cache. All user processes connected to the database share access to the buffer cache.
Shared Pool	The shared pool caches information that can be shared among users: ■ SQL statements that can be reused ■ Information from the data dictionary such as user account data, table and index descriptions, and privileges ■ Stored procedures, which is executable code that is stored in the database

Table 5–1 (Cont.) SGA Components

Component	Description
Redo Log Buffer	This buffer improves performance by caching redo information which is used for instance recovery until it can be written once at a more opportune time to the physical redo log files stored on disk. Redo information and redo log files are discussed in "Online Redo Log Files" on page 6-2.
Large Pool	This optional area is used to buffer large I/O requests for various server processes.
Java Pool	The Java pool memory is used for all session-specific Java code and data within the Java Virtual Machine (JVM)
Streams Pool	The Streams pool is used by the Oracle Streams product.

Program Global Area (PGA)

A program global area (PGA) is a memory area used by a single Oracle server process. A server process is a process that services a client's requests. Each server process has its own nonshared PGA when the process is started.

The PGA is used to process SQL statements and to hold logon and other session information.

The amount of PGA memory used and its content depends on the instance configuration, that is, whether the instance is running in dedicated server or shared server mode.

See Also: "Managing Memory Parameters" on page 5-9

Oracle Background Processes

Oracle creates a set of background processes for an instance to manage memory structure, asynchronously perform I/O to write data to disk, and do general maintenance tasks. The background processes consolidate functions that would otherwise be handled by multiple Oracle programs running for each user process. They monitor other Oracle processes to provide increased parallelism for better performance and reliability.

The background processes that are present depend on the features that are being used in the database. The most common background processes, and ones that most directly affect you, are described in Table 5–2, " Oracle Background Processes".

Table 5–2 Oracle Background Processes

Background Process	Description
Database Writer (DBW*n*)	The database writer writes modified blocks from the database buffer cache to the files on disk. Oracle allows a maximum of 20 database writer processes.
Log Writer (LGWR)	The log writer process writes redo log entries to disk. Redo log entries are generated in the redo log buffer of the SGA and the log writer process writes the redo log entries sequentially into an online redo log file.
Checkpoint	At specific times, all modified database buffers in the SGA are written to the datafiles by a database writer process (DBW*n*). This event is called a checkpoint. The checkpoint process signals DBW*n*, updates all of the datafiles and control files of the database, and records the time of this update.

Table 5–2 (Cont.) Oracle Background Processes

Background Process	Description
System Monitor (SMON)	The system monitor performs crash recovery when a failed instance starts up again.
Process Monitor (PMON)	The process monitor performs recovery when a user process fails. It cleans up the cache and frees resources that the failed process was using.
Archiver (ARC*n*)	Archiver processes copy the redo log files to archival storage when the log files are full or a log switch occurs. The database must be in archive log mode to run archive process. For more information, see Chapter 9, "Performing Backup and Recovery".

Accessing the Database

The database instance can be started only by an authorized database administrator with a special connection privilege to the Oracle instance. After the database instance has been started, it is usually open for access by users with database accounts.

About Database Administration Privileges

The administrator who starts the Oracle instance and database must connect to the instance with a special connection privilege. There are two of these privileges: SYSDBA for fully empowered database administrators and SYSOPER for users who operate the database, but have no privileges to access user objects. Authorization for these privileges is granted either by the operating system or by a special password file.

When you create an Oracle database, there are two primary administrative user accounts that are automatically created: SYS and SYSTEM. Both of these users have full database administration privileges, but only user SYS can initially connect with the SYSDBA privilege.

Starting the Oracle Database The process of starting the instance and database is as follows:

1. Start the instance using one of the following methods:

   - Use Oracle Enterprise Manager. See "Shutting Down and Restarting the Instance and Database with Enterprise Manager" on page 5-6.
   - Use the SQL*Plus STARTUP statement. See *Oracle Database Administrator's Guide*.
   - On Windows, you can start the Oracle services. See "Starting and Shutting Down the Database Instance on Windows" on page 5-6.

   Oracle reads the initialization parameter file, allocates SGA memory, and starts the background processes for the instance.

2. If you choose to mount the database, then the instance opens the database control file. The database is now said to be in the mount state. This state enables administrators to perform certain administrative functions that cannot be performed when other users are accessing the database.

3. If you choose to open the database, then the instance opens the redo log files and datafiles for the database. The database is now open and available for all user access.

The default startup behavior is to complete the three stages as described earlier. Unless you explicitly specified otherwise, the instance will be started, and the database mounted and opened.

Shutting Down the Oracle Database To shut down the database, reverse the start up procedure:

1. Shut down the database using one of the methods in "Starting the Oracle Database" on page 5-4. Any modified data blocks cached in the SGA that have not been written to disk are now written to disk. The redo log buffer is flushed. Datafiles are checkpointed and their headers are marked current as of the time the database was closed. Datafiles and log files are closed. Users can no longer access the database.
2. The Oracle instance dismounts the database and updates relevant entries in the control file to record a clean shutdown. The control file is closed. The database is now closed and only the instance remains.
3. The Oracle instance stops the background processes of the instance and deallocates the shared memory used by the SGA.

Server and Client Processes

In addition to background processes, Oracle creates server processes that handle the connection requests of user or client processes. A user connection is composed of two distinct pieces:

- A client program acting on behalf of the user, such as Oracle Enterprise Manager, SQL*Plus, or an application
- A server process that handles the connection to the database on behalf of the client program

When the client and Oracle operate on the same machine, you can combine the user process and corresponding server process into a single process to reduce system overhead. However, when the client and Oracle operate on different machines, a user process always communicates with Oracle through a separate server process.

Server processes can be either dedicated or shared. In dedicated server mode, each client process has its own server process. While a dedicated server process is good for long running queries and administrative tasks, an idle process or too many dedicated processes can result in an inefficient use of resources.

Using shared server mode eliminates the need for a dedicated server process for each connection. A dispatcher directs incoming network session requests to a pool of shared server processes. An idle shared server process picks up a request from a common queue, which means a small number of shared servers can perform the same amount of processing as many dedicated servers. Because the amount of memory required for each user is relatively small, less memory and process management are required, and more users can be supported.

Shared server mode is more efficient at supporting multiple users and clients making frequent short-running queries.

Network Connections

Oracle Net is a layer of software that allows different physical machines to communicate to access an Oracle database. Oracle Net runs on the client machine and on the database server. The client machine and the database server are often the same machine.

Oracle Net enables the client and server to communicate over a network using many popular network protocols, and it provides location transparency such that the client machine does not need to know the server's location. When the database is moved to another location, you only need to reconfigure Oracle Net. No changes are necessary to client applications.

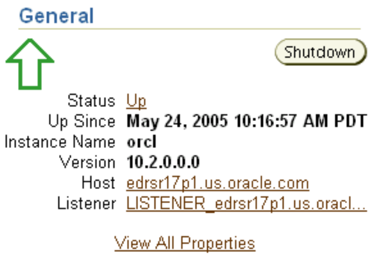
Oracle Net must be separately configured and started for it to handle client connections to the database. You can configure Oracle Net using Oracle Enterprise Manager or with a separately launched GUI tool called the Oracle Net Manager. To learn more, see Chapter 4, "Configuring the Network Environment".

When an instance starts, a listener process establishes a communication pathway to Oracle. When a user process makes a connection request, the listener determines whether it should use a shared server dispatcher process or a dedicated server process and establishes an appropriate connection.

Shutting Down and Restarting the Instance and Database with Enterprise Manager

If you need to shut down and later restart your instance or database, you can do this by clicking the **Shutdown** or **Startup** button, under the General heading on the Database Home page as shown in Figure 5–1, "Shutdown/Startup Button on Database Home Page".

Figure 5–1 Shutdown/Startup Button on Database Home Page



Status	Up
Up Since	May 24, 2005 10:16:57 AM PDT
Instance Name	orcl
Version	10.2.0.0.0
Host	edrsr17p1.us.oracle.com
Listener	LISTENER_edrsr17p1.us.oracl...

The first page to appear is Startup/Shutdown: Specify Host and Target Database Credentials. This page requires you to log in to the machine that Oracle is running on, and into the database itself with `SYSDBA` or `SYSOPER` privileges. The next screen enables you to shut down the database, or start up the database, specifying options.

Starting and Shutting Down the Database Instance on Windows

You can start and shut down your Oracle database with the Windows Services program. To start Oracle, you must start the following services:

- `OracleService`*`SID`*, which is your Oracle Database instance
- `Oracle`*`ORACLE_HOME`*`TNSListener`, which is your listener; required for clients to connect to your database
- `OracleDBConsole`*`SID`*, which enables clients to connect to Enterprise Manager

In the preceding service names, *`SID`* refers to the system identifier for the instance and *`ORACLE_HOME`* refers to the Oracle home.

To start these services:

1. Click **Start** and then select **Settings**, **Control Panel**, **Administrative Tools**, and **Services**.

   A list of all available services on your system appears.

2. Locate the Oracle services listed at the beginning of this section. For example, if your SID is `orcl`, then locate the following services:

   - `OracleServiceORCL`
   - `OracleOraDb10g_home1TNSListener`
   - `OracleDBConsoleorcl`

3. Start all three services. You can use the following techniques:

   - Right-click the service and **Start**.
   - In the **Action** menu, select **Start**.
   - Double-click the service to open a dialog box and then click **Start**.

To shut down the database, click **Stop** instead of **Start**.

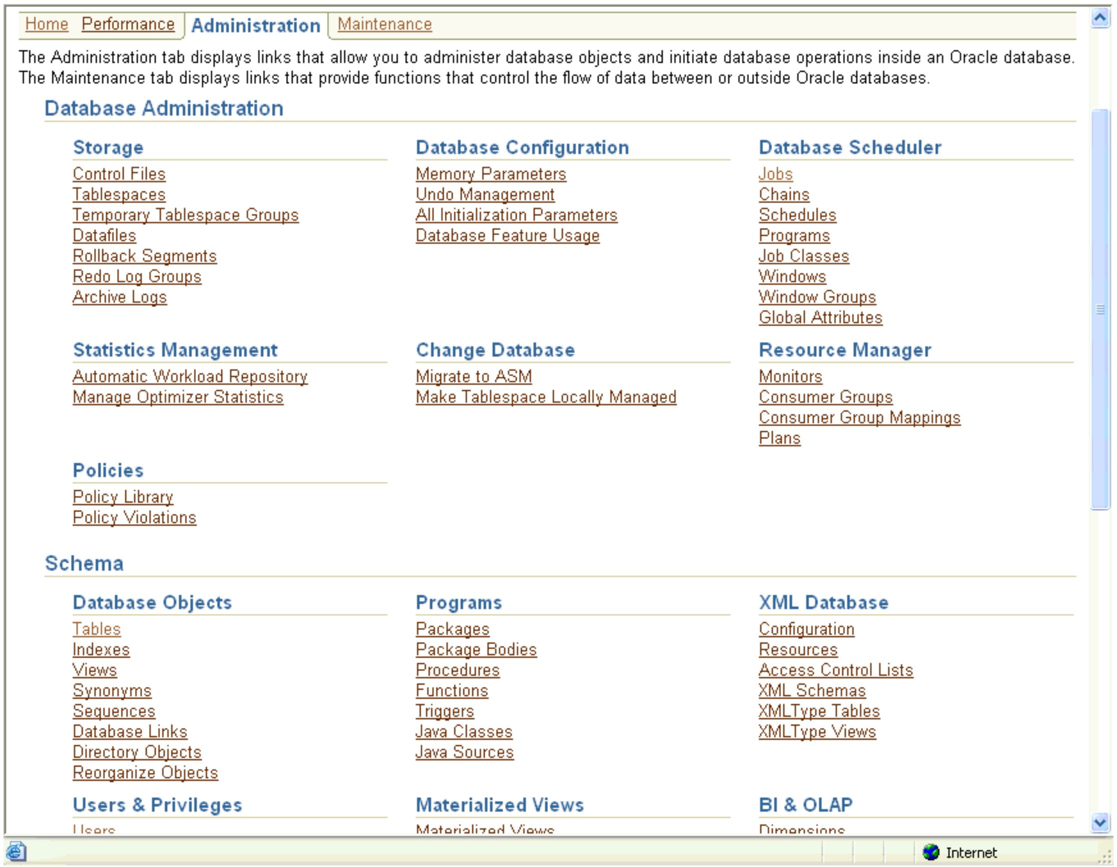
Viewing and Modifying Initialization Parameters

When you install one of the preconfigured databases provided by Oracle, the initialization parameters are optimized for normal use in the environment that you specified. The steps described in this section will familiarize you with viewing the initial parameter settings for your database and modifying parameters.

To view or modify the initialization parameters for your database:

1. From the Database Home page, click the **Administration** property page. The Administration home page appears as shown in Figure 5–2.

Figure 5–2 Database Administration Page



The Administration tab displays links that allow you to administer database objects and initiate database operations inside an Oracle database.
The Maintenance tab displays links that provide functions that control the flow of data between or outside Oracle databases.

Database Administration

Storage
Control Files
Tablespaces
Temporary Tablespace Groups
Datafiles
Rollback Segments
Redo Log Groups
Archive Logs

Database Configuration
Memory Parameters
Undo Management
All Initialization Parameters
Database Feature Usage

Database Scheduler
Jobs
Chains
Schedules
Programs
Job Classes
Windows
Window Groups
Global Attributes

Statistics Management
Automatic Workload Repository
Manage Optimizer Statistics

Change Database
Migrate to ASM
Make Tablespace Locally Managed

Resource Manager
Monitors
Consumer Groups
Consumer Group Mappings
Plans

Policies
Policy Library
Policy Violations

Schema

Database Objects
Tables
Indexes
Views
Synonyms
Sequences
Database Links
Directory Objects
Reorganize Objects

Programs
Packages
Package Bodies
Procedures
Functions
Triggers
Java Classes
Java Sources

XML Database
Configuration
Resources
Access Control Lists
XML Schemas
XMLType Tables
XMLType Views

Users & Privileges
Users

Materialized Views
Materialized Views

BI & OLAP
Dimensions

2. Under Database Configuration, click **All Initialization Parameters**.

   Enterprise Manager displays the Initialization Parameters page, where you can view the value of each initialization parameter for the database instance. You can use this page to alter initialization parameter values, using the **Show SQL**, **Revert**, and **Apply** buttons.

   There are two property pages shown on the Initialization Parameters page:

   - **Current**—This page displays all of the initialization parameter values that are currently in memory for the database instance.

     You can use this page to make dynamic changes to parameters in the current running instance. Only parameters marked dynamic can be changed. To do so, enter a new value and click **Apply**.

   - **SPFile**—This property page is present when you are using a server parameter file, whose location is displayed a the top of the table. This property page shows parameter settings in the server parameter file.

     Changes to parameter settings in this file are persistent across instance startup and shutdown. To make persistent changes to an initialization parameter, enter the new value and click **Apply**. You can optionally apply changes to the current running instance by checking **Apply changes in SPFile to the current running instance**. If you do not check this box, your changes will not take effect until the database is shut down and restarted.

Parameters marked **Basic** represent a small subset of all initialization parameters, and are needed to keep the database running smoothly.

As the number of database users grows and workload increases, you might need to alter some initialization parameters. You can make these changes using the Initialization Parameter page as described, or using one of the advisors provided by Oracle. One such advisor, the memory advisor, is described in "Managing Memory Parameters" on page 5-9.

Managing Memory Parameters

Memory parameters are initialization parameters that determine the total size of the system global area (SGA) and the program global area (PGA), and of the subcomponents of the SGA. The settings of memory parameters can affect the performance of your database. When you install your database, these parameters are tuned to meet the requirements of the environment that you specify.

If you enabled Automatic Shared Memory Configuration when you configured your database, Oracle automatically sizes the subcomponents of the SGA, which include the shared pool and buffer cache. Oracle recommends that you enable automatic memory tuning as such.

If Automatic Shared Memory Management is disabled, you can enable it on the Memory Parameters page. You can navigate to this page from the Administration page by clicking **Memory Parameters** under the Instance heading. From this page you can enable or disable Automatic Shared Memory Management and view your SGA and PGA memory settings from their property pages.

Modifying Memory Parameters

To modify the size of an SGA subcomponent without shutting down your instance, you can disable Automatic Shared Memory Management on the Memory Parameters page. With the Advice button on the Memory Parameters page, the Memory Advisor (discussed in "Using the Memory Advisor" in Chapter 10, "Monitoring and Tuning the Database") gives you advice on optimal memory settings. If you choose to modify your memory settings manually, you can enter new sizes and apply changes dynamically while the instance is up.

You can also modify memory parameters from the Current property page on the Initialization Parameters page. However, there are some restrictions on dynamic modification of memory parameters. For more information, see *Oracle Database Administrator's Guide*.

To make changes to memory parameters persistent across instance startup and shutdown, you must alter your parameter file. You can do so by navigating to SPFile property page from the Memory Parameters page and making your changes there.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information about memory parameters and their tuning

Instances: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter, and includes annotated screen shots.

To view the Instances OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/instance/instance.htm
```

6

Managing Database Storage Structures

This chapter discusses using Enterprise Manager to explore and manage the storage structures of your database. This chapter contains the following topics:

- Exploring the Storage Structure of Your Database
- Performing Common Database Storage Tasks
- Managing Undo for Your Database
- Making Changes to the Database Storage Structure
- Storage: Oracle by Example Series

Exploring the Storage Structure of Your Database

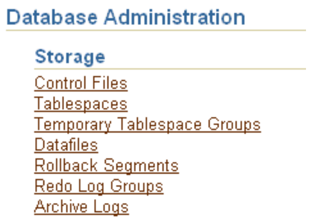
An Oracle database is made up of physical and logical structures. Physical structures are those that can be seen and operated on from the operating system, such as the physical files that store data on disk.

Logical structures are created and recognized by the Oracle database server and are not known to the operating system. The primary logical structure in a database, a tablespace, contains physical files. The applications developer or end user may be aware of the logical structure, but is not usually aware of this physical structure. The DBA must understand the relationship between the physical and logical structures of a database.

The Oracle database server can automate the management of its structure. Furthermore, Oracle Enterprise Manager provides a Web-based GUI to enable easy management and monitoring of your database.

To view database storage structure, navigate to the Storage section of the Administration Home page (see Figure 5–2, "Database Administration Page"). You can click the links shown in Figure 6–1 to access the storage pages.

Figure 6–1 Storage Options



The following sections provide a closer look at the database storage structure:

- Control Files
- Online Redo Log Files
- Archive Log Files
- Rollback Segments
- Tablespaces
- Datafiles
- Other Storage Structures

Control Files

A control file tracks the physical components of the database. It is essential to the functioning of the database. Because of the importance of the control file, Oracle recommends that the control file be multiplexed. In other words, the control file should have multiple identical copies. For databases created with DBCA, three copies of the control file are automatically created and kept in sync with each other.

If any control file fails, then your database becomes unavailable. As long as you have a control file copy, however, you can shut down your database and re-create the failed control file from the copy, then restart your database. Another option is to delete the failed control file from the `CONTROL_FILES` initialization parameter and restart your database with the remaining control files.

> **See Also:** *Oracle Database Administrator's Guide* for detailed information about control files

Displaying Control File Information

Click **Control Files** to open the Control Files general page. This page shows whether your database has a multiplexed control file. The Advanced and Record Section pages give you more detailed information about your control file. Explanations of this information are contained in the online Help.

Online Redo Log Files

Every Oracle database has a set of two or more online redo log files. The set of redo log files is collectively known as the redo log for the database. A redo log is made up of redo entries, which are also called redo records.

The redo log records all changes made to data. If a failure prevents modified data from being permanently written to the datafiles, then the changes can be obtained from the redo log, so work is never lost. To protect against a failure involving the redo log itself, Oracle multiplexes the redo log so that two or more identical copies of the online redo log can be maintained on different disks.

Your database's redo log consists of groups of redo log files. A group consists of a redo log file and its multiplexed copies. Each identical copy is said to be a member of that group. Each group is defined by a number, such as Group 1.

The database log writer process writes redo records from the memory buffer to a redo log group until the group fills up or a log switch operation is requested. The log writer then writes to the next group. The log writer performs this action in a circular fashion so that the oldest group is overwritten by the most recent redo records.

See Also: *Oracle Database Administrator's Guide* for detailed information about redo logs

Displaying Redo Log File Information

Click **Redo Log Groups** to open the Redo Log Groups page. This page shows the attributes of the redo log groups for your database. A database typically consists of three groups. Note the status attribute for the redo log groups. The `CURRENT` status group is the one currently being written to disk by the log writer.

When a redo log group contains only one member, it is not multiplexed. Click the number of a redo log group to view information about the redo log group members.

Multiplexing the Redo Log

While small database environments might not need multiplexed redo log groups, consider adding multiplexing as your database grows in users and applications. Multiplexing provides better protection for data in the case of instance or media failure.

To multiplex your redo log, you must add members to each redo log group. It is not required that redo log groups be symmetrical, but Oracle recommends that your groups all have the same number of members. A database must have a minimum of two redo log groups.

To create a multiplexed redo log for your database:

1. From the Storage section of the Administration page, click **Redo Log Groups**.

   The Redo Log Groups page appears.

2. Select a group and click **Edit**.

   The Edit Redo Log Group page appears.

3. In the **Redo Log Members** section, click **Add**.

   The Add Redo Log Member page appears.

4. Enter the file name for the new redo log member. For example, if your existing member file name is `REDO01.log`, then you might name this member `REDO01a.log`.

   Enter the file directory or accept the default. You can create this file in the same directory, but it is recommended that you store members on separate drives. That way, if there is a drive failure, you still have access to one member.

5. Click **Continue**, then Click **Apply** to accept your changes.

6. Repeat these steps for every existing group.

To switch a log file:

1. Navigate to the Redo Log Groups page.

2. From the **Actions** menu, select **Switch logfile**.

3. Click **Go**.

   You can see that the next groups status changes from `INACTIVE` to `CURRENT`.

Archive Log Files

When you archive your redo log, you write redo log files to another location prior to their being overwritten. This location is called the archive log. You can archive to multiple locations, including a standby database.

These copies of redo log files extend the amount of redo data that can be saved and used for recovery. Archiving can be either enabled or disabled for the database, but Oracle recommends that you enable archiving.

> **See Also:** *Oracle Database Administrator's Guide* for detailed information about archived redo logs

Displaying Archive Log File Information

Click **Archive Logs** to display the Archive Logs page. This page lets you view archive log files and their attributes.

For placing the database into archive log mode, or adding archive log locations, see "Configuring Your Database for Basic Backup and Recovery" on page 9-4.

Rollback Segments

Rollback segments were database structures used to track undo information for the database in earlier releases of Oracle. Now, the preferred way of managing undo is with the undo tablespace. For more information, see "Managing Undo for Your Database" in this chapter.

Displaying Rollback Segment Information

Click **Rollback Segments** to display the Rollback Segments page. This page displays the attributes of the system rollback segment. The system rollback segment is self-managing.

Datafiles

Datafiles are the operating system files that hold the data within the database. The data is written to these files in an Oracle proprietary format that cannot be read by other programs. Tempfiles are a special class of datafiles that are associated only with temporary tablespaces.

Datafiles can be broken down into the following components:

Segments and Extents

A segment contains a specific type of database object. For example, tables are stored in data segments, whereas indexes are stored in index segments.

An extent is a contiguous set of data blocks within a segment. Oracle initially allocates an extent of a specified size for a segment, but if that extent fills, then more extents can be allocated.

Data Blocks

Data blocks, also called database blocks, are the smallest unit of I/O to database storage. An extent consists of several contiguous data blocks. The database uses a default block size at database creation.

After the database has been created, it is not possible to change the default block size without re-creating the database. Nevertheless, it is possible to create tablespace with a block size different than the default block size. For details on how to create tablespaces with a nondefault or nonstandard block size, see *Oracle Database Administrator's Guide*.

Viewing Datafile Information

Click **Datafiles** to open the Datafiles page. You can use this page to view information about your datafiles. Select a datafile and click **View** to display more attributes for the datafile. You can also click the name of the tablespace to display the attributes of the owning tablespace.

> **See Also:** *Oracle Database Administrator's Guide* for detailed information about datafiles

Tablespaces

A database consists of one or more tablespaces. A tablespace is a logical structure, consisting of one or more datafiles or tempfiles.

Tablespaces hold data in the form of other objects such as tables and indexes. When you create an Oracle database, some tablespaces already exist, such as SYSTEM and USERS. For a complete list, see "Some Tablespaces in the Database" on page 6-6.

You can create new tablespaces to support your user and application data requirements. During tablespace creation, you set the following parameters:

- Locally-Managed vs. Dictionary-Managed Tablespaces
- Tablespace Type
- Tablespace Status
- Auto-Extend Tablespace

Locally-Managed vs. Dictionary-Managed Tablespaces

Space management within a tablespace involves keeping track of free and used space, so that space is allocated efficiently during data insertion and deletion.

Oracle recommends creating locally-managed tablespaces over dictionary-managed tablespaces. Locally-managed tablespaces keep the space allocation information within the tablespace, not in the data-dictionary, thus offering better performance.

By default, Oracle sets all newly created tablespaces to be locally-managed with automatic segment management, a feature that further improves performance.

Tablespace Type

Tablespaces can be one of three types: permanent, undo, or temporary.

Permanent Tablespaces

Oracle uses these tablespaces to store permanent data, such as system data. You use permanent tablespaces to store your user and application data.

Undo Tablespace

A database running in automatic undo management mode transparently creates and manages undo data in the undo tablespace. Oracle uses undo data to provide read-consistency and to enable features such as Flashback query.

Even though you can create more than one undo tablespace, only one can be active. If you want to switch the undo tablespace used by the database instance, however, then you can create a new one and instruct the database to use it instead. The undo tablespace no longer in use can be dropped later.

Temporary Tablespaces

Temporary tablespaces are used for storing temporary data, as would be created when SQL statements perform sorts. After database creation, the database already has a temporary tablespace.

You would create another temporary tablespace if you were creating a temporary tablespace group. Under normal circumstances, you need not create additional temporary tablespaces.

Temporary and permanent tablespaces can be assigned at the user level. You can create a default temporary tablespace at database creation time. This tablespace is used as the default temporary tablespace for users who are not otherwise assigned a temporary tablespace.

> **See Also:** *Oracle Database Administrator's Guide* to learn more about temporary tablespaces

Tablespace Status

You can set tablespace read-write status and its availability as follows:

Read Write

Users can read and write to the tablespace after it is created. This is the default.

Read Only

If the tablespace is created read-only, then the tablespace cannot be written to until its status is changed to Read Write. It is unlikely that you create a read-only tablespace, but you might change it to that status after you have written data to it that you do not want modified.

Offline

If the tablespace is created offline, then no users can access it. It is unlikely that you create an offline tablespace, but you might change its status to offline later to perform maintenance on its underlying files.

Auto-Extend Tablespace

You can set a tablespace to automatically extend itself by a specified amount when it reaches its limit.

If you do not enable autoextend, you will be alerted when the tablespace reaches its critical or warning threshold size. The critical and warning threshold parameters have default values which you can change at any time. You can respond to space alerts by manually increasing the tablespace.

Some Tablespaces in the Database

Table 6–1 describes some of the tablespaces included in the database.

Table 6–1 Tablespaces and Descriptions

Tablespace	Description
EXAMPLE	This tablespace contains the sample schemas that Oracle includes with the database. The sample schemas provide a common platform for examples. Oracle documentation and educational materials contain examples based on the sample schemas.
SYSAUX	This is an auxiliary tablespace to the SYSTEM tablespace. Some components and products that used the SYSTEM tablespace or their own tablespaces prior to Oracle Database 10*g* now use the SYSAUX tablespace. Using SYSAUX reduces the load on the SYSTEM tablespace and reduces maintenance because there are fewer tablespaces to monitor and maintain. Every Oracle Database 10*g* or higher level database must have a SYSAUX tablespace. Components that use SYSAUX as their default tablespace during installation include Automatic Workload Repository, Oracle Streams, Oracle Text, and Enterprise Manager Repository. For more information, see the *Oracle Database Administrator's Guide*.
SYSTEM	This tablespace is always created at database creation. Oracle uses it to manage the database. It contains the data dictionary, which is the central set of tables and views used as a read-only reference for a particular database. It also contains various tables and views that contain administrative information about the database. These are all contained in the SYS schema, and can only be accessed by user SYS or other administrative users with the required privilege.
TEMP	This tablespace stores temporary data generated when processing SQL statements. For example, this tablespace would be used for query sorting. Every database should have a temporary tablespace that is assigned to users as their temporary tablespace. In the preconfigured database, the TEMP tablespace is specified as the default temporary tablespace. This means that if no temporary tablespace is specified when a user account is created, then Oracle assigns this tablespace to the user.
UNDOTBS1	This is the undo tablespace used by the database server to store undo information. See "Managing Undo for Your Database" on page 6-14 to understand how an Oracle database uses the undo tablespace. Every database must have an undo tablespace.
USERS	This tablespace is used to store permanent user objects and data. Like the TEMP tablespace, every database should have a tablespace for permanent user data that is assigned to users. Otherwise, user objects will be created in the SYSTEM tablespace, which is not good practice. In the preconfigured database, USERS is assigned the default tablespace, and space for all objects created by non-system users comes from this tablespace. For system users, the default permanent tablespace remains SYSTEM.

Other Storage Structures

Other storage structures that can exist in an Oracle database include the initialization parameter file, the password file, and backup files.

Initialization Parameter File

Initialization parameters are used by the database server at startup to determine the runtime resources for the database. They are actively monitored by the database and can be set or modified while the database is running.

Initialization parameters and the initialization parameter file are discussed in Chapter 5, "Managing the Oracle Instance".

Password File

A database can use a password file to authenticate administrative users with SYSDBA connect privileges. SYSDBA privileges enable a DBA to start up and shut down the database and perform other high-level administrative tasks. This password file is outside of the database itself because it must sometimes be referenced when the database is not yet running.

This is not the only form of administrator authentication, so not all databases require a password file.

The password file is discussed in "Accessing the Database" in Chapter 5, "Managing the Oracle Instance".

Backup Files

Backup files are not technically database files, but are copies of the database in some form that can be used to recover the database if a failure causes loss of data.

Backup files are discussed in Chapter 9, "Performing Backup and Recovery".

Performing Common Database Storage Tasks

As a DBA, you must understand how to perform the following tasks related to managing storage:

- Viewing Tablespaces in Your Database
- Creating a Tablespace
- Modifying a Tablespace
- Dropping a Tablespace
- Reclaiming Wasted Space

Viewing Tablespaces in Your Database

You can view all tablespaces in your database. From the Storage section of the Administration page, click **Tablespaces** to open the Tablespaces page. This page shows the following information about each tablespace:

- Type of tablespace
- Type of extent management
- Type of segment management
- Status
- Size
- Percent of tablespace in use

Click the tablespace name for more information.

Creating a Tablespace

A user, groups of users, or an application might require its own tablespace for creating schema objects, rather than using the default USER tablespace. To create a tablespace, follow the steps listed in this section.

1. From Storage section of the Administration page, click **Tablespaces**.

   The Tablespaces page appears.

2. Click **Create** to open the Create Tablespace General page.

   Alternatively, to create a tablespace that is like an existing tablespace, select an existing tablespace. From the **Actions** list, select **Create Like** and then click **Go**. The other options that are available from the **Actions** menu are shortcuts for various tablespace operations.

3. Enter a name for the tablespace. For example, enter `TB1`.

4. In the Extent Management section, select **Locally Managed**. See "Locally-Managed vs. Dictionary-Managed Tablespaces" on page 6-5.

5. In the Type section, select **Permanent**. See "Tablespace Type" on page 6-5.

6. In the Status section, select **Read Write** from among the status options. See "Tablespace Status" on page 6-6.

7. In the Datafiles section of the page, click **Add** to add one or more datafiles to the tablespace. See "Datafiles" on page 6-4.

   > **Note:** If you check **Use bigfile tablespace**, then the tablespace can have only one datafile. Bigfile tablespaces are used with very large databases that use Automatic Storage Management or other logical volume managers that support striping, RAID, and dynamically extensible logical volumes.

8. In the Add Datafiles page, enter a file name. For example, enter `TB1.dbf`. Accept the **File Directory** and **File Size** defaults.

9. In the Storage section, select **Automatically extend datafile when full (AUTOEXTEND)** to allow the Oracle database to automatically expand the datafile. Specify an amount in the **Increment** field by which you want to extend the datafile each time it fills. Leave the Maximum File Size set to **Unlimited**. Click **Continue** to return to the Create Tablespace page.

10. Click **Storage** to open the Create Tablespace Storage property page. Leave all of the defaults settings on the Storage page. See online Help for more information on these settings.

11. Click **OK** to add the tablespace. You can now see the new tablespace in the Tablespaces page.

Modifying a Tablespace

With Enterprise Manager, you can modify a tablespace. For example, you can extend it, set it to automatically extend, change its space thresholds, or change its status to offline. When you get a critical and warning alert, you might need to extend a tablespace or take it offline to recover it.

This section shows you how to set a tablespace to automatically extend when it reaches its limit.

1. From the Storage section of the Administration page, click **Tablespaces**.

   The Tablespaces page appears.

2. Select a tablespace to autoextend, such as `USERS`, and click **Edit**.

   The Edit Tablespace: USERS General property page appears.

3. Select the datafile associated with the tablespace and click **Edit**.

The Edit Datafile page appears.

4. Check **Automatically extend datafile when full (AUTOEXTEND)**. Set a suitable increment, such as 1 MB. You can optionally set a maximum file size or set file size to unlimited depending on your resources.
5. Click **Continue** to return to the Edit Tablespace page.
6. Click **Apply**.

   A confirmation message appears.

Similarly, you can change tablespace thresholds by selecting a tablespace on the Tablespaces page and clicking **Edit**. Click **Thresholds** to navigate to the Threshold property page. This page lets you set monitored thresholds for space usage. You receive a warning when the threshold is reached. You can accept the default, specify different thresholds, or disable threshold warnings. See Chapter 10, "Monitoring and Tuning the Database" for more information about monitoring the database.

You can take a tablespace offline by selecting a tablespace on the Tablespaces page and clicking **Edit**. Select **Offline** on the Edit Tablespace General page and then click **Apply**.

Dropping a Tablespace

After a tablespace has been dropped, the objects and data in it are no longer available. To recover them can be a time-consuming process. Oracle recommends performing a backup before and after dropping a tablespace.

1. From the Storage section on the Administration page, click **Tablespaces**.

   The Tablespaces page appears.
2. Select the tablespace that you want to drop and click **Delete**.

   Enterprise Manager asks for confirmation that you want to delete the tablespace and associated datafiles from the operating system.
3. Click **Yes** to remove the tablespace.

Reclaiming Wasted Space

Over time, inserts, updates, and deletes (DML operations) to objects within a tablespace can create pockets of empty space that individually are not big enough to be reused. Collectively, these pockets can combine to form large amounts of wasted space. The resulting sparsely-populated objects can suffer performance degradation during queries and DML operations.

An Oracle database enables you reclaim wasted space in the following ways:

Shrink Operation

Data is compacted to the front of the table. Free space can either be returned to the tablespace or kept in the table for future inserts. A shrink operation on a table does not affect DML operations to the table during the operation.

Reorganization Operation

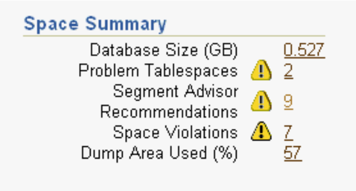
The object is re-created in a different part of the tablespace. This operation has the desirable side effect of compacting the object. For this operation to succeed, however, the tablespace must already have free space equal to the size of the object.

Viewing Segment Advisor Recommendations

The Segment Advisor identifies objects such as tablespaces and tables that have unused space you can reclaim. It performs its analysis by examining usage and growth statistics and by sampling the data in the object. It is configured to run automatically at regular intervals, and you can also run it manually.

The regularly scheduled Segment Advisor run is known as the Automatic Segment Advisor. Results from the advisor are summarized on the Space Summary section of the home page as **Segment Advisor Recommendations**. See Figure 6–2, "Home Page Space Summary". Click the adjacent link to open the Segment Advisor Recommendations page.

Figure 6–2 Home Page Space Summary



Space Summary

Database Size (GB)	0.527
Problem Tablespaces	⚠ 2
Segment Advisor Recommendations	⚠ 9
Space Violations	⚠ 7
Dump Area Used (%)	57

Objects such as tablespaces, tables, and indexes are made up of one or more segments. The Segment Advisor page summarizes space usage, estimating the amount of reclaimable space for each tablespace. From this page you can drill down to the segment level. See Figure 6–3, "Segment Advisor Recommendations".

Recommendations can be either a shrink or reorganization operation. If you created the tablespace with automatic segment space management, which is the default, then the Segment Advisor recommends shrinking. If the segment is not eligible for shrink, or if the tablespace was created with manual segment space management, then the Segment Advisor recommends other reorganizing methods.

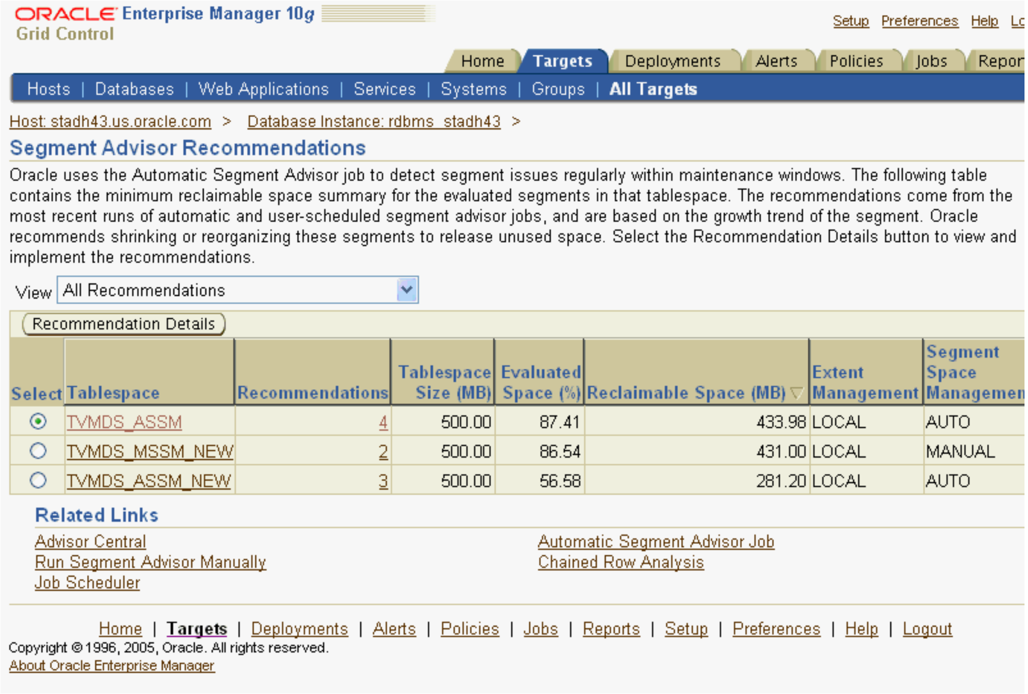
The following section describes how to view the recommendations and then shrink your objects using the shrink wizard.

To view recommendations and shrink an object:

1. From the Space Summary section of the Administration page, click the link adjacent to **Segment Advisor Recommendations**. Figure 6–2, "Home Page Space Summary" shows the link.

   The Segment Advisor Recommendations page appears, as shown in Figure 6–3, "Segment Advisor Recommendations".

Figure 6–3 Segment Advisor Recommendations


ORACLE Enterprise Manager 10g
Grid Control
Setup Preferences Help Lo
Home Targets Deployments Alerts Policies Jobs Repor
Hosts | Databases | Web Applications | Services | Systems | Groups | All Targets

Host: stadh43.us.oracle.com > Database Instance: rdbms_stadh43 >

Segment Advisor Recommendations

Oracle uses the Automatic Segment Advisor job to detect segment issues regularly within maintenance windows. The following table contains the minimum reclaimable space summary for the evaluated segments in that tablespace. The recommendations come from the most recent runs of automatic and user-scheduled segment advisor jobs, and are based on the growth trend of the segment. Oracle recommends shrinking or reorganizing these segments to release unused space. Select the Recommendation Details button to view and implement the recommendations.

View: All Recommendations

Select	Tablespace	Recommendations	Tablespace Size (MB)	Evaluated Space (%)	Reclaimable Space (MB) ▽	Extent Management	Segment Space Managemen
●	TVMDS_ASSM	4	500.00	87.41	433.98	LOCAL	AUTO
○	TVMDS_MSSM_NEW	2	500.00	86.54	431.00	LOCAL	MANUAL
○	TVMDS_ASSM_NEW	3	500.00	56.58	281.20	LOCAL	AUTO

Related Links

Advisor Central
Run Segment Advisor Manually
Job Scheduler
Automatic Segment Advisor Job
Chained Row Analysis

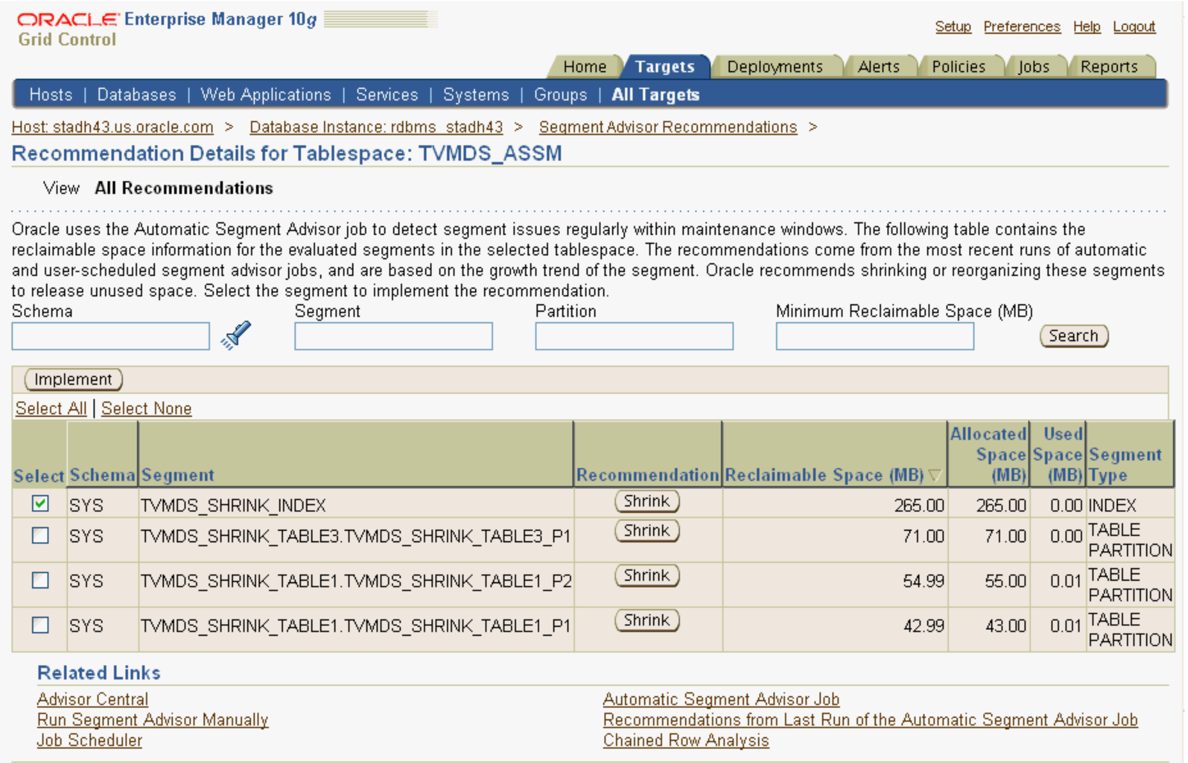
Home | Targets | Deployments | Alerts | Policies | Jobs | Reports | Setup | Preferences | Help | Logout

Copyright © 1996, 2005, Oracle. All rights reserved.
About Oracle Enterprise Manager

2. Select a tablespace and then click **Recommendation Details**.

   The Recommendation Details for Tablespace page appears, as shown in Figure 6–4, "Segment Advisor Recommendation Details". This page lists the tablespace segments that are eligible for shrinking.

Figure 6–4 Segment Advisor Recommendation Details



ORACLE Enterprise Manager 10g
Grid Control
Setup Preferences Help Logout
Home | Targets | Deployments | Alerts | Policies | Jobs | Reports
Hosts | Databases | Web Applications | Services | Systems | Groups | All Targets
Host: stadh43.us.oracle.com > Database Instance: rdbms_stadh43 > Segment Advisor Recommendations >

Recommendation Details for Tablespace: TVMDS_ASSM

View **All Recommendations**

Oracle uses the Automatic Segment Advisor job to detect segment issues regularly within maintenance windows. The following table contains the reclaimable space information for the evaluated segments in the selected tablespace. The recommendations come from the most recent runs of automatic and user-scheduled segment advisor jobs, and are based on the growth trend of the segment. Oracle recommends shrinking or reorganizing these segments to release unused space. Select the segment to implement the recommendation.

Schema: Segment: Partition: Minimum Reclaimable Space (MB): (Search)

(Implement)

Select All | Select None

Select	Schema	Segment	Recommendation	Reclaimable Space (MB) ▽	Allocated Space (MB)	Used Space (MB)	Segment Type
☑	SYS	TVMDS_SHRINK_INDEX	(Shrink)	265.00	265.00	0.00	INDEX
☐	SYS	TVMDS_SHRINK_TABLE3.TVMDS_SHRINK_TABLE3_P1	(Shrink)	71.00	71.00	0.00	TABLE PARTITION
☐	SYS	TVMDS_SHRINK_TABLE1.TVMDS_SHRINK_TABLE1_P2	(Shrink)	54.99	55.00	0.01	TABLE PARTITION
☐	SYS	TVMDS_SHRINK_TABLE1.TVMDS_SHRINK_TABLE1_P1	(Shrink)	42.99	43.00	0.01	TABLE PARTITION

Related Links

Advisor Central
Run Segment Advisor Manually
Job Scheduler
Automatic Segment Advisor Job
Recommendations from Last Run of the Automatic Segment Advisor Job
Chained Row Analysis

3. Select a segment to shrink by using either of the following techniques:

   - Click **Shrink** next to the segment you want to shrink.
   - Select one or more segments and click **Implement**.

   The Shrink Segment: Options page appears.

   You can also reorganize an object at this step. To do so, click the **Reorg** button and continue with the reorganization wizard.

4. Accept **Compact Segments and Release Space**. This action returns freed space to the tablespace. If you do not want to release the freed space to the tablespace, then choose **Compact Segments**. Click **Implement**.

   The Shrink Segment: Schedule page appears.

5. Note your job name. Under Start, select to run the job **Immediately**. The shrink operation can be resource intensive, so Oracle recommends running it during off-peak hours. You can also select **Later** and schedule a better time to run the operation. Click **Submit**.

The scheduler Jobs page appears and shows the status of the job.

6. Click **Refresh** to update the page until the status of the job is Completed. You can also click View Job Status.

Running the Segment Advisor Manually

You can run the Segment Advisor manually. You can do so when you want to analyze objects not selected by the Automatic Segment Advisor, or when you want more up-to-date recommendations on a tablespace.

See Also: *Oracle Database Administrator's Guide* for more information on running the Segment Advisor

Managing Undo for Your Database

This section discusses undo management. These tasks involve storing the changes of database transactions long enough to accommodate rollback, read consistency, and flashback features.

After you install the database, you can start building your database without immediately managing undo because the database automatically performs this task. Later, as your database activity and transaction rate increase, understanding how to manage undo becomes more useful to you.

This section contains the following topics:

- About Undo Data
- Undo Tablespace Size and Retention Time
- Automatic Undo Management
- Managing Undo with Enterprise Manager
- Using the Undo Advisor

About Undo Data

When a transaction modifies the database, Oracle copies the original data before modifying it. The original copy of the modified data is called undo data. Saving this information is necessary for the following reasons:

- To undo any uncommitted changes made to the database in the event that a rollback operation is necessary. A rollback operation can be the result of a user who wants to undo the changes of a misguided or unintentional transaction, or it can be part of a recovery operation.
- To provide read consistency, which means that each user can get a consistent view of data, even while other uncommitted changes may be occurring against the data. For example, if a user issues a query at 10:00 a.m. and the query lasts for 15 minutes, then the query results should reflect the entire state of the data at 10:00 a.m., regardless of updates or inserts by other users during the query.
- To enable certain Flashback features, namely Flashback Query and Flashback Table, which enable you to view or recover data to a previous point in time.

Undo Tablespace Size and Retention Time

Undo data is stored in a logical database structure called an undo tablespace. The undo tablespace is of finite size, so records might be overwritten as transactions occur.

Oracle saves undo data at least until the transaction has been committed. Until this time, the undo data is in the active state. Therefore, the amount of space available in the undo tablespace should be at least large enough to hold the active undo data generated by current transactions. Otherwise, some of these transactions might fail. When active undo data is stored in the undo tablespace, Oracle automatically ensures that it is never overwritten until the corresponding transaction has been committed.

Even after the transaction has been committed, the undo data still cannot be overwritten immediately to ensure the success of Flashback functionality, and for read consistency for long running transactions. For example, if your longest query takes 15 minutes, then the undo tablespace should be large enough to hold 15 minutes worth of undo data.

To control the retention of undo records, Oracle maintains an undo retention period. This period indicates the amount of time that must pass before Oracle overwrites undo data. The undo retention period affects the size of the undo tablespace; the longer the retention period, the more space is needed.

The undo retention period should be at least as long as your longest-running query. By default, Oracle automatically extends the undo tablespace to accommodate the longest-running query based on your ongoing system activity. Nevertheless, you might need to manually increase the size of your undo tablespace in the following circumstances:

- When your undo tablespace is set to a fixed size (auto-extend disabled) and long running queries are failing with snapshot too old errors.
- When you plan to use Flashback features to recover from user errors such as unintentional changes. In this case, the undo retention should be set equal to the period between the present and the earliest point in time to which you want to return. For more details on Flashback features, see *Oracle Database Administrator's Guide*.

Automatic Undo Management

The Oracle database automatically determines how long undo data should be kept based on the time your queries take to run. Undo data preserved within this window of time is said to be in the unexpired state. After this time, the state of the undo data changes to expired. Undo data is a good candidate for overwriting only when it is in the expired state.

The length of time that Oracle keeps undo data in the unexpired state depends on your tablespace configuration. When you create your database with DBCA, the undo tablespace is set by default to automatically extend itself to maintain unexpired undo for the longest-running query.

With a fixed-sized undo tablespace, Oracle automatically keeps the undo data in the unexpired state for the longest possible time for the tablespace of the specified size. If the undo tablespace does not have adequate free or expired space to store active undo data generated by current transactions, however, then Oracle might be forced to overwrite the unexpired undo data. This situation might cause long-running queries to fail with an error and an alert.

To avoid situations in which long-running queries can fail, it is recommended that you let Oracle automatically extend the size of the undo tablespace. By default, the undo tablespace is set to auto-extend when you use DBCA to configure your database.

You may choose to disable auto-extension and adjust the size of the tablespace manually. In this case, ensure that the tablespace is large enough to meet the read-consistency requirements for your longest-running query. Also, if you use Flashback features, then the tablespace must be large enough to accommodate Flashback operations. Oracle Enterprise Manager includes an Undo Advisor to help you determine the optimal size. See "Using the Undo Advisor" on page 6-17.

Managing Undo with Enterprise Manager

With Enterprise Manager, you can manage undo as follows:

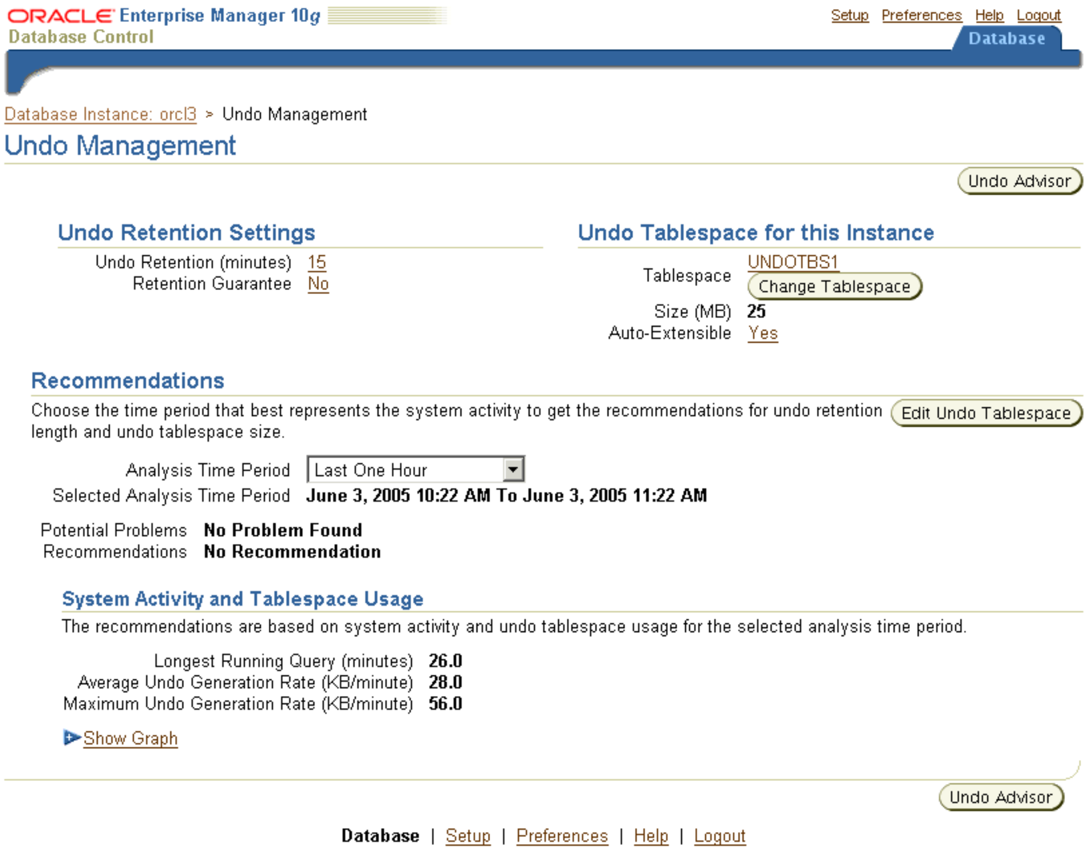
1. From the Database Control home page, click **Administration**.

   The Administration property page appears.

2. In the Database Configuration section, click **Undo Management**.

   The Undo Management page appears, as shown in Figure 6–5.

Figure 6–5 Undo Management page



Database Instance: orcl3 > Undo Management

Undo Management

Undo Retention Settings

Undo Retention (minutes) 15
Retention Guarantee No

Undo Tablespace for this Instance

Tablespace UNDOTBS1
Size (MB) **25**
Auto-Extensible Yes

Recommendations

Choose the time period that best represents the system activity to get the recommendations for undo retention length and undo tablespace size.

Analysis Time Period Last One Hour
Selected Analysis Time Period **June 3, 2005 10:22 AM To June 3, 2005 11:22 AM**

Potential Problems **No Problem Found**
Recommendations **No Recommendation**

System Activity and Tablespace Usage

The recommendations are based on system activity and undo tablespace usage for the selected analysis time period.

Longest Running Query (minutes) **26.0**
Average Undo Generation Rate (KB/minute) **28.0**
Maximum Undo Generation Rate (KB/minute) **56.0**

You can use the Undo Management page to view the following about your undo configuration:

- Name and size of undo tablespace
- Auto-extend tablespace setting

- Auto-tuned undo retention period
- Minimum retention period

In Undo Tablespace for this Instance, the **Auto-Extensible** field shows **Yes** if auto-extending the tablespace is enabled, which is the default. When the undo tablespace is auto-extensible, Oracle automatically increases the size of the tablespace when more space is needed. By combining automatic extension of the undo tablespace with automatically tuned undo retention, you can ensure that long-running queries succeed by guaranteeing the undo for such queries.

If you have a fixed-size tablespace, then you can use this page as a starting point for determining space requirements and extending the tablespace. See "Using the Undo Advisor" on page 6-17.

The Undo Retention Settings section describes your minimum undo retention period. While Oracle automatically tunes the undo retention period, minimum undo retention enables you to define the lowest value allowable for your database. When you create a database, the minimum undo retention is set to a default value. You might need to alter this value to build a recovery strategy using Flashback Query. See "Setting Minimum Undo Retention Time" on page 6-20.

The Recommendations section of the Undo Management page gives you recommendations on undo tablespace size based on your system activity. Recommendations and alerts can arise when you are using a fixed-size tablespace and queries have been failing because of insufficient undo space. On rarer occasions, when using an auto-extend tablespace, the same recommendations can arise if the system has reached it maximum disk limit.

If Oracle recommends that you extend the undo tablespace, then you can use the Undo Advisor to determine a better size.

Using the Undo Advisor

The amount of undo data that can be retained depends on the size of your undo tablespace. If your tablespace is set to auto-extend, then Oracle automatically acquires space as needed. If you choose to disable auto-extend, however, then you are responsible for ensuring that the undo tablespace has enough space. In this situation, Oracle configures the undo retention to the maximum possible value for that tablespace size.

The Undo Advisor helps you analyze various scenarios to determine an appropriate undo tablespace size for different values of maximum undo retention. This analysis might be required in the following situations:

- You have a fixed-sized tablespace (auto-extend disabled) and the auto-tuned value of undo retention is not large enough to prevent queries from failing. The Undo Advisor can help you determine a better tablespace size to ensure successful completion of your queries. See "Gaining Advice from Undo Advisor" on page 6-18.
- You use Flashback features such as Flashback Query or Flashback Table. For Flashback operations to go back in time, the database must ensure that undo data is not overwritten.

  To build a flashback recovery strategy, you can set the minimum undo retention, which determines the lowest value for automatic undo tuning. For example, if the low threshold is set to 15 minutes, then Oracle never lowers the undo retention time to less than 15 minutes. Consequently, if your flashback recovery strategy requires you to go back 8 hours to recover from human errors, then set the minimum undo retention to 8 hours. To learn how to set a new retention time, see "Setting Minimum Undo Retention Time" on page 6-20

Gaining Advice from Undo Advisor

The Undo Advisor can help you determine a better size for your undo tablespace to ensure successful completion of queries. Follow these general steps to determine the required new size of your tablespace:

1. Determine the duration of your longest running query according to your application characteristics. If this duration is longer than the tuned retention, then these long-running queries will encounter an error. In this case, your undo tablespace is too small. You need to either set your tablespace to auto-extend or manually extend it.
2. In the Database Configuration section of the Administration page, click **Undo Management**.

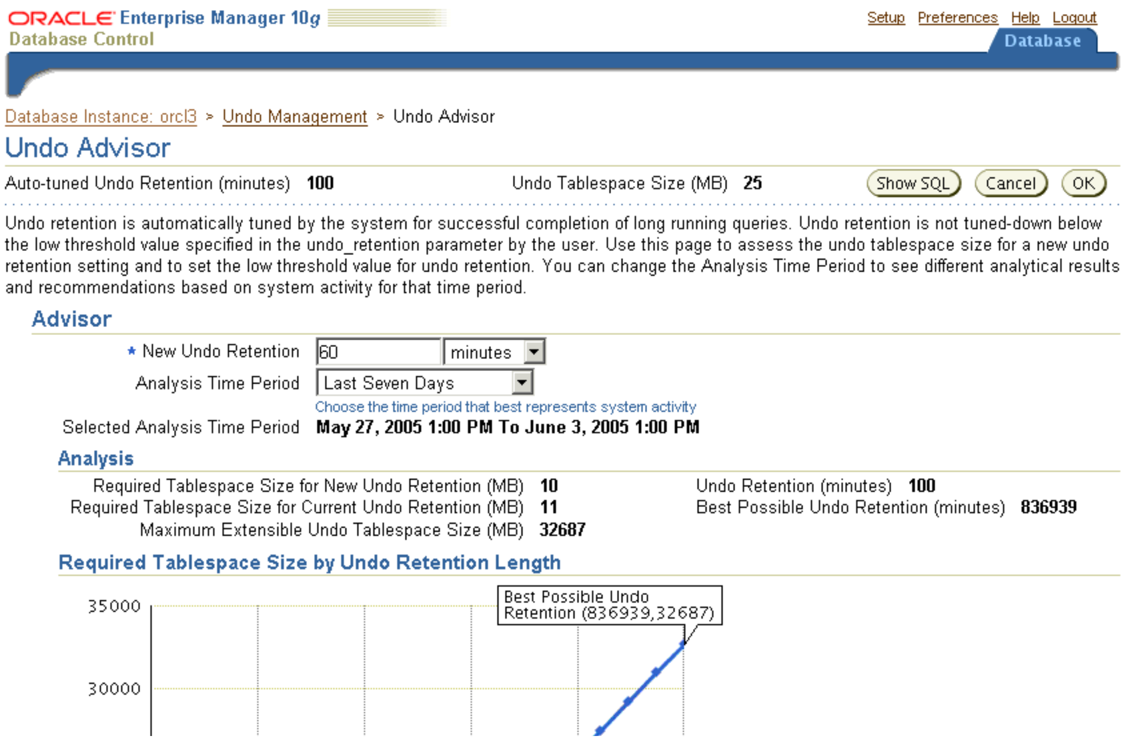
   The Undo Management page appears, as shown in Figure 6–5.
3. Click **Undo Advisor**.

   The Undo Advisor page appears. The top of the page shows the current auto-tuned undo retention time and undo tablespace size, as shown in Figure 6–6.
4. In the **New Undo Retention** field of the Advisor section, plug in the value of your longest-running query.
5. In the **Analysis Time Period** list, select a time period that best reflects your business cycle.

   The Analysis section displays the Required Tablespace size for New Undo Retention.

For example, suppose we have a fixed size tablespace of 25 MB. We determine the longest running query is 60 minutes. We enter this number in the **New Undo Retention** field as shown in Figure 6–6. The advisor recommends 10 MB of space to accommodate this query. Because our tablespace is 25 MB, our undo tablespace is adequately configured.

Figure 6–6 Undo Advisor with Fixed-Sized Tablespace



ORACLE Enterprise Manager 10g
Database Control
Setup Preferences Help Logout
Database
Show SQL
Cancel
OK
35000
30000
Best Possible Undo Retention (836939,32687)

Database Instance: orcl3 > Undo Management > Undo Advisor

Undo Advisor

Auto-tuned Undo Retention (minutes) **100**

Undo Tablespace Size (MB) **25**

Undo retention is automatically tuned by the system for successful completion of long running queries. Undo retention is not tuned-down below the low threshold value specified in the undo_retention parameter by the user. Use this page to assess the undo tablespace size for a new undo retention setting and to set the low threshold value for undo retention. You can change the Analysis Time Period to see different analytical results and recommendations based on system activity for that time period.

Advisor

* New Undo Retention 60 minutes

Analysis Time Period Last Seven Days

Choose the time period that best represents system activity

Selected Analysis Time Period **May 27, 2005 1:00 PM To June 3, 2005 1:00 PM**

Analysis

Required Tablespace Size for New Undo Retention (MB) **10**

Required Tablespace Size for Current Undo Retention (MB) **11**

Maximum Extensible Undo Tablespace Size (MB) **32687**

Undo Retention (minutes) **100**

Best Possible Undo Retention (minutes) **836939**

Required Tablespace Size by Undo Retention Length

Additionally, the Required Tablespace Size by Undo Retention Length graph shows the relationship between retention period and undo tablespace size, highlighting key data points such as the Auto-tuned Undo Retention and Best Possible Undo Retention.

Extending the Undo Tablespace

When auto-extend tablespace is enabled, the system automatically extends the undo tablespace if it is under space pressure. When the auto-extend tablespace feature is disabled, however, you might need to manually extend the undo tablespace. You might discover this information ahead of time while planning with the Undo Advisor. See "Using the Undo Advisor" on page 6-17.

You might also need to extend the undo tablespace when you get an undo tablespace alert (warning or critical), or when you get a *query too long* alert or *snapshot too old* error.

To resize the tablespace:

1. From the Undo Management page, click **Edit Undo Tablespace**.

   The Edit Tablespace page appears.

2. Select a datafile to extend and click **Edit**.

   The Edit Datafile page appears.

3. In the **File Size** field, enter a new datafile size.

   You can also have the system automatically extend the datafile by enabling **Automatically extend datafile when full** under Storage and specifying an increment size.

4. Click **Continue**.

   The Edit Tablespace page appears.

5. Under Datafiles, Click **Apply**.

   A confirmation message should appear.

Setting Minimum Undo Retention Time

If you are using the Flashback Query or Flashback Table feature and need to configure a flashback recovery strategy to go back in time, then configure the minimum undo retention as follows:

1. In the **New Undo Retention** field of the Undo Advisor page, enter the new retention period.

   For configuring Flashback Query and Flashback Table, set a value equal to how far back in time you need to go. For example, if you need an eight hour flashback recovery strategy, set the minimum retention to 8 hours.

2. Click **OK**.

Alternatively, you can set this parameter as you do for other system parameters by navigating to the All Initialization Parameters page. Set the parameter called `undo_retention`. For more information about this page, see "Viewing and Modifying Initialization Parameters" on page 5-7.

Making Changes to the Database Storage Structure

The preconfigured database that you installed includes all of the database structures of a basic database. As the user base grows, you can expand existing database storage structures or create additional ones. For example, you might need to create additional tablespaces for users or applications, or you might want to create additional online redo log groups to expand the redo log capacity.

Oracle provides alerts, advisories, and monitoring pages to help you make decisions regarding database storage. These topics are discussed in Chapter 10, "Monitoring and Tuning the Database". Oracle lets you create your own alerts for other events that you want to monitor.

The Storage section links shown in "Storage Options" on page 6-1 are links that you can use to make changes to the database storage structure. Other pages also contain links that enable you to perform specific actions that affect the database storage structure.

For example, the Maintenance page includes a Backup/Recovery Settings section. This section includes a link for changing the archiving mode of the database and creating archived redo log files (see "Configuring Your Database for Basic Backup and Recovery" on page 9-4).

When you make changes to the storage structure of your database, these changes can be reflected in the data dictionary and in the control file. It is important to consider this fact in your backup and recovery strategy. This topic is discussed in Chapter 9, "Performing Backup and Recovery".

Storage: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter and includes annotated screen shots.

To view the Storage OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/storage/storage.htm
```

7

Administering Users and Security

This chapter describes how to create and manage user accounts. It contains the following topics:

- Overview of Users and Security
- Administering Roles
- Administering Profiles
- Administering Database Users
- Users: Oracle by Example Series

Overview of Users and Security

For users to access your database, you the database administrator must create user accounts and grant appropriate database access privileges to those accounts. A user account is identified by a username and defines the user's security attributes, including the following:

- Authentication method
- Password (encrypted) for database authentication
- Tablespace access
- Tablespace quotas
- Whether the account is locked or not

User accounts are usually created by a database administrator or a person specifically assigned to create accounts and administer security. After you create a user account, you must grant basic privileges to the account to enable the user to connect to the database and to view and create database objects, known as schema objects. The name of the schema is identical to the name of the user.

All databases include the `SYS`, `SYSTEM`, `SYSMAN`, and `DBSNMP` administrative accounts. Other accounts are included depending upon what features or options are installed.

The administrative accounts provided by Oracle Database should be used only by authorized individuals. To protect these accounts from unauthorized access, these accounts are initially locked with their passwords expired. As the database administrator, you are responsible for the unlocking and resetting of these accounts as described in "Unlocking Accounts and Resetting Passwords" on page 7-11. Table 7–1 contains descriptions of some of these accounts.

Table 7–1 Administrative User Accounts Provided by Oracle Database

Username	Password	Description	See Also
CTXSYS	CTXSYS	The Oracle Text account	*Oracle Text Reference*
DBSNMP	DBSNMP	The account used by the Management Agent component of Oracle Enterprise Manager to monitor and manage the database	*Oracle Enterprise Manager Grid Control Installation and Basic Configuration*
MDDATA	MDDATA	The schema used by Oracle Spatial for storing Geocoder and router data	*Oracle Spatial User's Guide and Reference*
MDSYS	MDSYS	The Oracle Spatial and Oracle *inter*Media Locator administrator account	*Oracle Spatial User's Guide and Reference*
DMSYS	DMSYS	The data mining account. `DMSYS` performs data mining operations.	*Oracle Data Mining Administrator's Guide* *Oracle Data Mining Concepts*
OLAPSYS	MANAGER	The account used to create OLAP metadata structures. This account owns the OLAP Catalog (CWMLite).	*Oracle OLAP Application Developer's Guide*
ORDPLUGINS	ORDPLUGINS	The Oracle *inter*Media user. Plugins supplied by Oracle and third party format plugins are installed in this schema.	*Oracle interMedia User's Guide*
ORDSYS	ORDSYS	The Oracle *inter*Media administrator account	*Oracle interMedia User's Guide*
OUTLN	OUTLN	The account that supports plan stability. Plan stability enables you to maintain the same execution plans for the same SQL statements. `OUTLN` acts as a role to centrally manage metadata associated with stored outlines.	*Oracle Database Performance Tuning Guide*
SI_INFORMTN_SCHEMA	SI_INFORMTN_SCHEMA	The account that stores the information views for the SQL/MM Still Image Standard	*Oracle interMedia User's Guide*
SYS	CHANGE_ON_INSTALL	The account used to perform database administration tasks	"About Administrative Accounts" on page 7-4
SYSMAN	CHANGE_ON_INSTALL	The account used to perform Oracle Enterprise Manager database administration tasks. Note that `SYS` and `SYSTEM` can also perform these tasks.	*Oracle Enterprise Manager Grid Control Installation and Basic Configuration*
SYSTEM	MANAGER	Another account used to perform database administration tasks	"About Administrative Accounts" on page 7-4

Sample Schemas

Most of the included accounts are administrative accounts, but Sample Schema accounts are also present. The Oracle Database Sample Schemas are a set of interlinked schemas that enable Oracle documentation, and Oracle by Example Series, to illustrate common database tasks:

- The human resources (`hr`) schema is useful for introducing basic topics. An extension to this schema supports Oracle Internet Directory demos.
- The order entry (`oe`) schema is useful for dealing with matters of intermediate complexity. Many datatypes are available in this schema, including nonscalar datatypes.
- The online catalog (`oc`) subschema is a collection of object-relational database objects built inside the `oe` schema.
- The product media (`pm`) schema is dedicated to multimedia datatypes.

- The information exchange (ix) schemas demonstrate Oracle Advanced Queuing capabilities.
- The sales history (sh) schema is designed for demos with large amounts of data. An extension to this schema provides support for advanced analytic processing.

See Also: *Oracle Database Sample Schemas* for a description of the Sample Schemas that are used for examples in Oracle Database documentation and educational materials.

Administering Roles

System privileges, object privileges, and roles provide a basic level of database security. They are designed to control user access to data and to limit the kinds of SQL statements that users can execute.

About Privileges and Roles

Roles are groupings of privileges that you can use to create different levels of database access. For example, you can create a role for application developers that enable users to create tables and programs.

You can grant privileges and roles to other users only when you possess the necessary privilege. The granting of roles and privileges starts at the administrator level. At database creation, the administrative user SYS is created and granted all system privileges and predefined Oracle roles. User SYS can then grant privileges and roles to other users and also grant those users the right to grant specific privileges to others.

Table 7–2 provides descriptions and examples of privileges and roles.

Table 7–2 Privileges and Roles

Privilege or Role	Description	Examples
System privilege	An Oracle-defined privilege usually granted only to and by administrators. System privileges enable users to perform specific database operations.	The following are examples of system privileges that can be granted to users: ■ CREATE TABLE allows grantee to create tables in the grantee's schema. ■ CREATE USER allows grantee to create users in the database. ■ CREATE SESSION allows grantee to connect to an Oracle database to create a user session.
Object privilege	A privilege that controls access to a specific object.	The following examples are object privileges that can be granted to users: ■ SELECT ON hr.employees TO myuser ■ INSERT ON hr.employees TO myuser
Role	A group of privileges or other roles	The following examples are Oracle-defined roles: ■ CONNECT is a role that Enterprise Manager automatically grants to a user when you create a user as shown in "Creating Users" on page 7-8. This role has the CREATE SESSION privilege. ■ RESOURCE extends the privileges of a user beyond those granted by the CONNECT role. It includes CREATE PROCEDURE, CREATE TRIGGER, and other system privileges. ■ DBA is the standard role that can be granted by an administrator to another administrator. It includes all system privileges and should only be granted to the most trusted and qualified of users. Assigning this role to a user enables the user to administer the database. You can create your own roles if you have been granted this privilege.

See Also: *Oracle Database SQL Reference* for a list of Oracle-defined privileges and roles

About Administrative Accounts

The following administrative accounts are automatically created when Oracle Database is installed:

- SYS
- SYSTEM

SYS

When you create an Oracle database, the user SYS is automatically created and granted the DBA role.

All base tables and views for the database data dictionary are stored in the schema SYS. These base tables and views are critical for the operation of Oracle Database. To maintain the integrity of the data dictionary, tables in the SYS schema are manipulated only by the database. They should never be modified by any user or database administrator. Also, you should not create any tables in the schema of user SYS, although you can change the storage parameters of the data dictionary settings if necessary.

Ensure that most database users are never able to connect to Oracle Database with the SYS account.

SYSTEM

When you create an Oracle Database, the user SYSTEM is also automatically created and granted the DBA role.

The SYSTEM user can create additional tables and views that display administrative information as well as internal tables and views used by various Oracle Database options and tools. Never use the SYSTEM schema to store tables of interest to nonadministrative users.

A predefined DBA role is automatically created with every Oracle Database installation. This role contains most database system privileges. Therefore, you should grant the DBA role only to actual database administrators. The DBA role does not include the SYSDBA or SYSOPER system privileges.

Administrative Privileges

SYSDBA and SYSOPER are administrative privileges required to perform basic database operations such as creating the database and instance startup and shutdown. Depending upon the level of authorization you require, you must have one of these privileges granted to you.

> **Note:** The SYSDBA and SYSOPER system privileges allow access to a database instance even when the database is not open. Control of these privileges is totally outside of the database itself.

You can also think of the SYSDBA and SYSOPER privileges as types of connections that enable you to perform certain database operations for which privileges cannot be granted in any other way. For example, if you have the SYSDBA privilege, then you can connect to the database by specifying CONNECT AS SYSDBA.

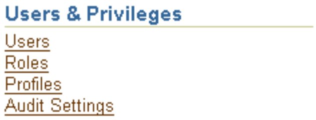
> **See Also:** *Oracle Database Administrator's Guide* for more the operations authorized with each privilege and an example

Viewing Roles

You can use Enterprise Manager to view existing roles as follows:

1. In the Users & Privileges section of the Administration home page, click **Roles**.

Figure 7–1 Users & Privileges



The Roles page appears. From this page you can create, edit, view, or delete roles. The structure and functionality of the Roles page is similar to that of the Users page shown in Figure 7–2.

2. Select the CONNECT role.

3. Click **View**.

The View page appears. In this page you can see all of the privileges and roles associated with the CONNECT role.

Creating Roles

You can create a secure role with the privileges necessary for application development. You can then grant the role to other roles or users depending on the level of data access required by the user.

See Also: *Oracle Database Security Guide* for more information on administering user security, roles, and privileges

In this exercise, you create an application developer role called APPDEV.

To create the APPDEV role:

1. In the Users & Privileges section of the Administration home page, click **Roles**.

   The Roles page appears.

2. Click **Create**.

   The Create Role General page appears.

3. In the **Name** field, enter the name of the new role. For instance, enter APPDEV to create a new role for application developers.

4. Click **OK**.

   A page appears with a list of all roles, including the APPDEV role that you just created. You can now modify this new role by adding the required privileges.

Modifying Roles

You can add roles, privileges, and consumer groups to roles. In this exercise, you add the basic system privileges shown in Table 7–3, which allow the creation of various objects, to the APPDEV role that you created previously. These objects are described in Chapter 8, "Managing Schema Objects".

Table 7–3 APPDEV Privileges

Privilege	Description
CREATE TABLE	Enables user to create tables in his schema.
CREATE VIEW	Enables user to create views in his schema.
CREATE PROCEDURE	Enables user to create procedures in his schema.
CREATE TRIGGER	Enables user to create triggers in his schema.
CREATE SEQUENCE	Enables user to create sequences in his schema.
CREATE SYNONYM	Enables user to create synonyms in his schema.

To modify the APPDEV role:

1. In the Users & Privileges section of the Administration home page, click **Roles**.

   The Roles page appears.

2. From the list of roles, select **APPDEV** and click **Edit**.

3. Click **System Privileges** to navigate to the System Privileges property page.

The System Privilege column should display no items.

4. Click **Edit List**.

   The Modify System Privileges page appears.

5. In the **Available System Privileges** list, double-click the privileges listed in Table 7–3 to add them to the **Selected System Privileges** list.

6. Click **OK**.

   You are returned to the Edit Role: APPDEV page.

7. Click **Apply**.

   A confirmation message should appear saying that the role has been modified successfully.

Dropping Roles

In this exercise, you drop to the APPDEV role that you created in "Dropping Roles" on page 7-7.

To drop the APPDEV role:

1. In the Users & Privileges section of the Administration home page, click **Roles**.

   The Roles page appears.

2. Select the APPDEV role and click **Delete**.

   A confirmation page appears.

3. Click **Yes**.

   A confirmation message indicates that the role has been deleted successfully.

Administering Profiles

A user profile establishes the password management policy for a user and sets limits the user's access to certain database resources. When you create the user in "Creating Users" on page 7-8, you assign the Oracle-supplied default profile. This default profile is liberal in its resource specifications and does not provide tight restrictions on password usage.

To display the attributes of the default profile:

1. In the Users & Privileges section of the Administration home page, click **Profiles**.

   The Profiles page appears. On this page you can create, edit, view, or delete profiles. The structure and functionality of the Profiles page is similar to that of the Users page shown in Figure 7–2.

2. Select the DEFAULT profile and click **View**.

   The View page appears. In this page you can view all of the attributes associated with the DEFAULT profile.

Database resource usage and limits are managed by the Database Resource Manager. You can read about the Database Resource manager in online Help and view its pages when you click the links in the Resource Manager section of the Database Administration page.

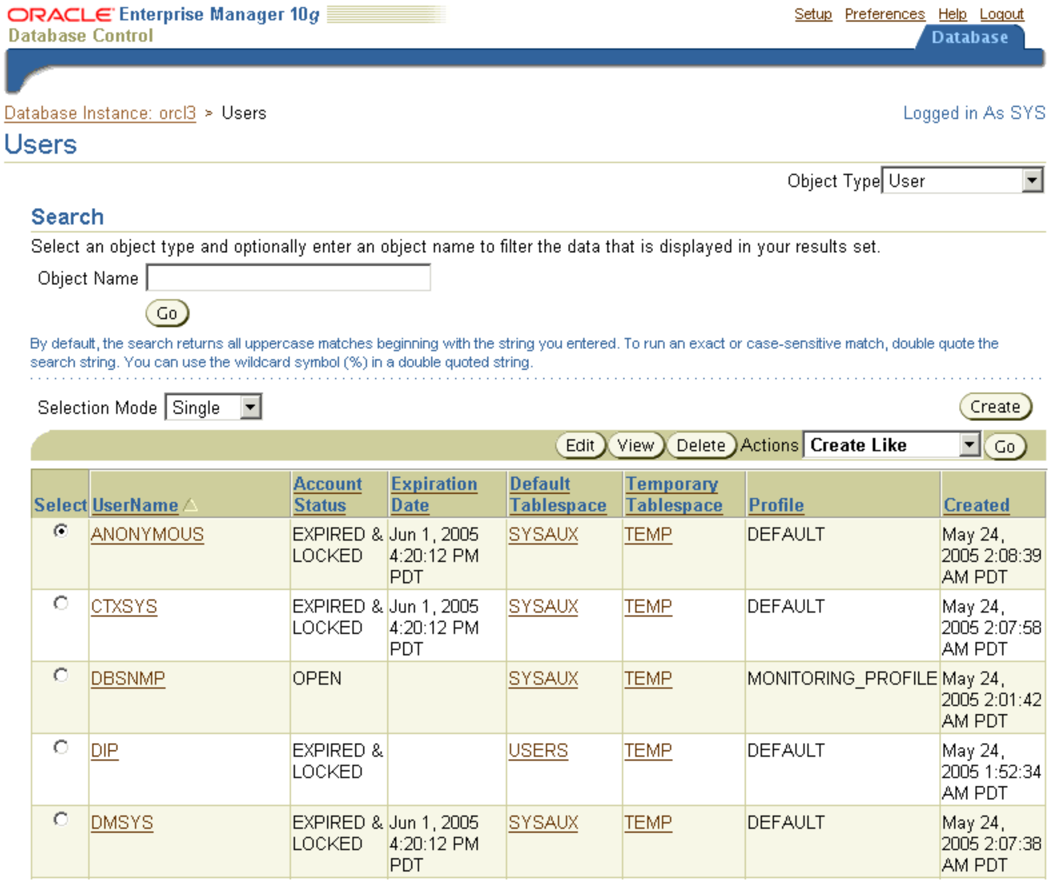
> **See Also:** *Oracle Database Administrator's Guide* as well as the Database Resource Manager online Help for a description of the Database Resource Manager and its use

Administering Database Users

You can use Enterprise Manager to administer database users. You can create specific users to enable access to the database based on roles and privileges. For example, you might create a specific user MYUSER with the necessary privileges (which you can assign to roles) to develop applications. You might create other users specifically to administer the database.

In the Users & Privileges section of the Administration home page, click **Users**. The Users page appears, as shown in Figure 7–2. From this page you can view, add, edit, or delete database users.

Figure 7–2 Users Page



Database Instance: orcl3 > Users

Logged in As SYS

Users

Object Type User

Search

Select an object type and optionally enter an object name to filter the data that is displayed in your results set.

Object Name

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode Single

Select	UserName	Account Status	Expiration Date	Default Tablespace	Temporary Tablespace	Profile	Created
●	ANONYMOUS	EXPIRED & LOCKED	Jun 1, 2005 4:20:12 PM PDT	SYSAUX	TEMP	DEFAULT	May 24, 2005 2:08:39 AM PDT
○	CTXSYS	EXPIRED & LOCKED	Jun 1, 2005 4:20:12 PM PDT	SYSAUX	TEMP	DEFAULT	May 24, 2005 2:07:58 AM PDT
○	DBSNMP	OPEN		SYSAUX	TEMP	MONITORING_PROFILE	May 24, 2005 2:01:42 AM PDT
○	DIP	EXPIRED & LOCKED		USERS	TEMP	DEFAULT	May 24, 2005 1:52:34 AM PDT
○	DMSYS	EXPIRED & LOCKED	Jun 1, 2005 4:20:12 PM PDT	SYSAUX	TEMP	DEFAULT	May 24, 2005 2:07:38 AM PDT

Creating Users

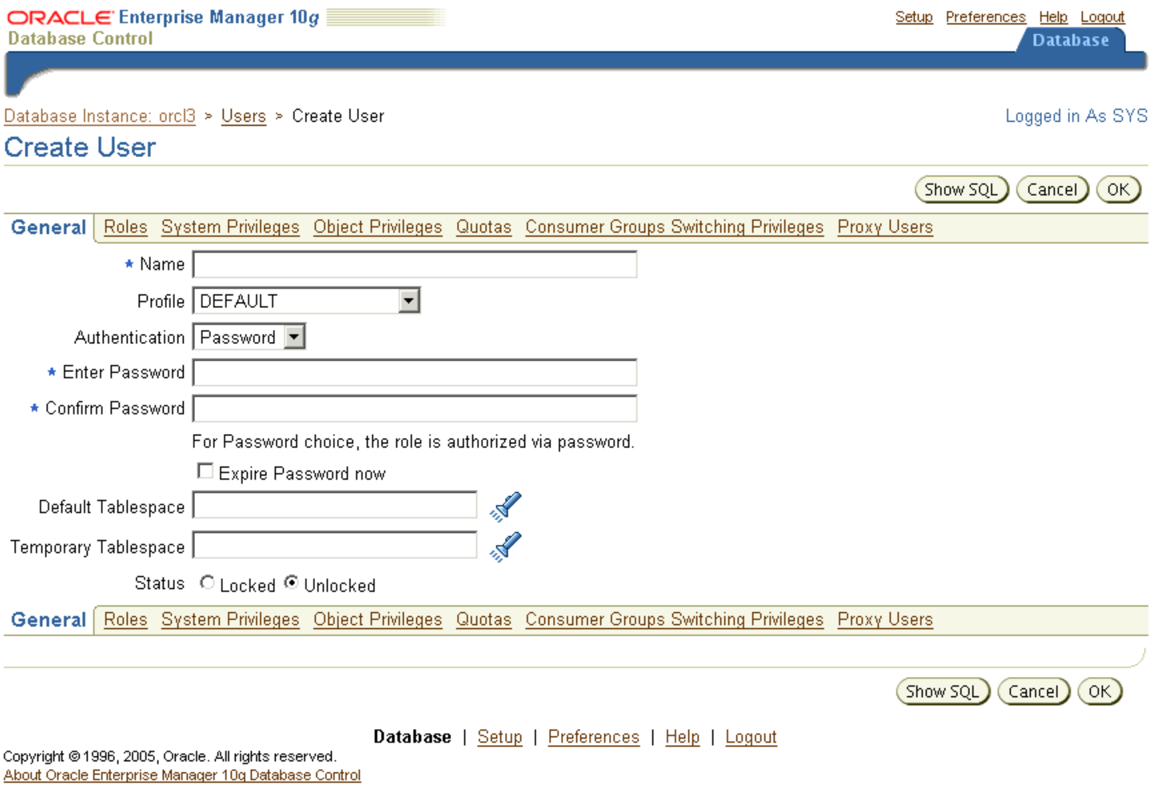
In this section, you create a user named `MYUSER`, set the password, and assign `MYUSER` to the `USERS` tablespace. This series of tasks enables `myuser` to log in to the database. Later, you learn how to edit and assign roles to this user.

To create a database user, perform the following tasks as user `SYS` or `SYSTEM`:

1. On the Users page, click **Create**.

The Create Users General page shown in Figure 7–3 is displayed. The General page is one of a series of pages in the Users property page.

Figure 7–3 Create User General Page



ORACLE Enterprise Manager 10g
Database Control
Setup Preferences Help Logout
Database
Database Instance: orcl3 > Users > Create User
Logged in As SYS
Show SQL
Cancel
OK
General
Roles
System Privileges
Object Privileges
Quotas
Consumer Groups Switching Privileges
Proxy Users

Create User

* Name	
Profile	DEFAULT
Authentication	Password
* Enter Password	
* Confirm Password	
	For Password choice, the role is authorized via password.
	□ Expire Password now
Default Tablespace	
Temporary Tablespace	
Status	○ Locked ● Unlocked

General
Roles
System Privileges
Object Privileges
Quotas
Consumer Groups Switching Privileges
Proxy Users
Show SQL
Cancel
OK
Database | Setup | Preferences | Help | Logout

About Oracle Enterprise Manager 10g Database Control

2. In the **Name** box, enter a user name. For example, enter `MYUSER`.

3. In the **Profile** list, accept the value `DEFAULT`. The profile specifies the resource limit.

   Note that this profile does not allow the user to log in to Enterprise Manager, which would require the user to be assigned the `DBA` role. For more information, see "Administering Roles" on page 7-3.

4. In the **Authentication** list, accept the value `Password`. For advanced authentication schemes, see *Oracle Database Security Guide*.

5. In the **Enter Password** and **Confirm Password** boxes, enter your password.

6. Do not check **Expire Password** now. If the account status to set to expired, then the user or the database administrator must change the password before the user can log in to the database.

7. For the **Default Tablespace** field, click the flashlight icon and select the `USERS` tablespace. While users can specify different defaults for each user, it is easiest to define a default permanent tablespace and a default temporary tablespace at the database level, instead of the user level. For more information on the `USERS` tablespace, see "Some Tablespaces in the Database" on page 6-6.

8. For the **Temporary Tablespace** field, click the flashlight icon and select the `TEMP` tablespace. For more information on the `TEMP` tablespace, see "Some Tablespaces in the Database" on page 6-6.

9. In **Status**, select **Unlocked**.

   You can later lock an account to keep a user out of the database. Locking an account is preferable to deleting a user, which removes all associated tables and data.

10. Click **OK**.

    The Users page appears. You can now see an entry for the new user `myuser`.

The Users General page has links to several other property pages. For example, you can specify user roles, privileges, quotas, consumer groups, and proxy users. Consumer groups are groups of users, or sessions, that are grouped together based on their processing needs.

> **Note:** This task creates a regular database user. To grant the user system administrator privileges, complete additional steps described in Chapter 3, "Getting Started with Oracle Enterprise Manager".

> **See Also:** *Oracle Database Administrator's Guide* and *Oracle Database Security Guide* for more information on roles, privileges, quotas, consumer groups and proxy authentication

Using a Shortcut to Create Users

If you later want to create other users similar to user `MYUSER`, or another existing user, Oracle provides a shortcut.

To use a shortcut to create a new user:

1. Select the user that you want to duplicate. For example, select `MYUSER`.
2. From the **Actions** menu, select **Create Like**.
3. Click **Go**.

   The Create User General page appears. This page displays a new user with the same attributes as the duplicated user.

The **Actions** list also provides shortcuts for other actions, as well as providing a means to display the SQL DDL used to create a user.

Editing Users

You can change the user attributes by navigating to the Users page (Figure 7–2) and clicking **Edit**. You can then select the properties page with the attributes that you want to change.

Changing the Tablespace Quota for a User

When you create a new user, the user is not able to save data until given a tablespace quota. This procedure enables user `MYUSER` to save data.

To change the tablespace quotas assigned to user `MYUSER`:

1. From the Users page, select `MYUSER` in the results list, then click **Edit**.

   The Edit User General page appears.

2. Click **Quotas** to display the Quotas property page.

This user has 0 MB quota on all tablespaces, which means that MYUSER has no quota in any tablespace. Because MYUSER belongs to the USERS tablespace, he must have quota to create tables and other schema objects in this tablespace.

3. In the **Quota** list for tablespace USERS, select **Value**.
4. In the corresponding **Value** column, enter 100.
5. Click **Apply**.

   User MYUSER can now create objects in the USERS tablespace.
6. Click the **Database** tab to return to the Administration home page.

Unlocking Accounts and Resetting Passwords

Locked accounts cannot be accessed by the user. During installation, you can unlock and reset the Oracle-supplied database user accounts. If you did not choose to unlock those accounts at that time, then you can do so now.

To unlock database account:

1. From the Users page shown in Figure 7–2, select a user whose **Account Status** is shown as EXPIRED AND LOCKED. For example, select DMSYS.
2. From the **Actions** list, select **Unlock User** and click **Go**.
3. Click **Yes** to the confirm that you want to unlock the user.

   This action unlocks the user account. The **Account Status** is now EXPIRED. This action does not reset the password, so the user is still unable to log in to the database.
4. From the Users page, select the same user and click **Edit**.

   The Edit User General page appears.
5. Enter a password for the account and click **Apply**.

You must follow the preceding steps individually for each account that you want to unlock and reset.

Locking an account is similar to unlocking it. You select the user and then choose **Lock User** from the **Actions** list. Locking an account denies access to the account.

To better understand the ramifications of unlocking and resetting accounts, see the other property pages available on the Edit user page. Specifically, click the **Roles**, **System Privileges**, and **Object Privileges** links to see the privileges of the user whose account you are enabling.

Granting Roles

You can use Enterprise Manager to grant roles to users. For example, you can grant a user the DBA role, which allows the user to administer the database. You can also grant user-created roles, such as the APPDEV role you created in "Creating Roles" on page 7-6.

To grant the DBA and APPDEV role to MYUSER:

1. In the Users page, select MYUSER and click **Edit**.

   The Edit User: MYUSER page appears.
2. Click **Roles**.

   A page appears with the list of roles for MYUSER. The only role should be CONNECT.

3. Click **Edit List**.

   The Modify Roles page appears.

4. In the **Available Roles** list, select the DBA and click **Move** to add it to the **Selected Roles** list. Do the same for the APPDEV role. You can also move the role by double-clicking.

5. Click **OK**.

   The Roles property page appears.

6. Click **Apply** to save your changes.

Revoking Roles

You can also revoke roles from users.

To revoke the DBA role from MYUSER:

1. In the Users page, select MYUSER and click **Edit**.

   The Edit User: MYUSER page appears.

2. Click **Roles**.

   A page appears with the list of roles for MYUSER.

3. Click **Edit List**.

   The Modify Roles page appears.

4. In the **Selected Roles** list, select the DBA role and click **Remove** to make it part of the **Available Roles** list.

5. Click **OK**.

   The Roles property page.

6. Click **Apply** to save your changes.

Dropping Users

Enterprise Manager enables you to drop users. You must exercise caution when dropping users, however, because this action drops all schema objects owned by the user including tables and indexes.To deny user access to the database, it is better to lock the user account or expire the user password. See "Unlocking Accounts and Resetting Passwords" on page 7-11.

To drop MYUSER:

1. In the Users page, select MYUSER.

2. Click **Delete**.

   A confirmation page appears.

3. Click **Yes** to confirm the deletion.

Users: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter and includes annotated screen shots.

To view the Users OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/users/users.htm
```

8

Managing Schema Objects

This chapter discusses the management of schema objects. The chapter contains the following sections:

- Overview of Schema Objects
- Managing Tables
- Managing Indexes
- Managing Views
- Managing Database-Resident Program Units
- Working with Other Schema Objects
- Schemas: Oracle by Example Series

Overview of Schema Objects

A schema is a collection of database objects. A schema is owned by a database user and has the same name as that user. Schema objects are logical structures created by users. Objects such as tables or indexes hold data, or can consist of a definition only, such as a view or synonym.

Note: There is no relationship between a tablespace and a schema. Objects in the same schema can use storage in different tablespaces, and a tablespace can contain data from different schemas.

You can create and manipulate schema objects with SQL or with Oracle Enterprise Manager. The underlying SQL is generated for you by Oracle Enterprise Manager.

This section contains the following topics:

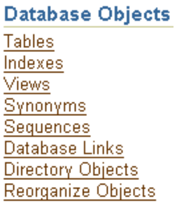
- Exploring Database Schema Objects
- Naming Schema Objects

See Also: *Oracle Database SQL Reference* for more detailed information about schema objects, object names, and data types

Exploring Database Schema Objects

To explore and manage the schema objects in your database, open the Database Administration property page (see Figure 5–2, "Database Administration Page") and click one of the links in the Database Objects section. Figure 8–1 shows these links.

Figure 8–1 Database Objects Links



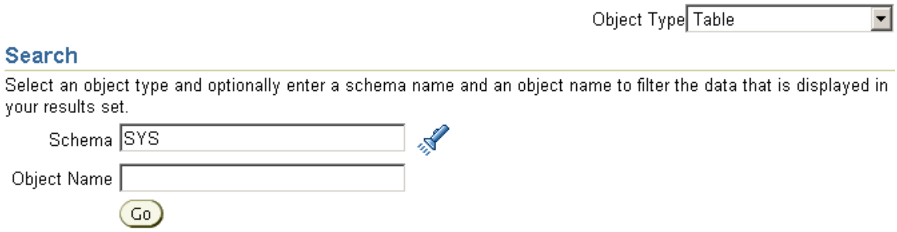
Database Objects

Tables
Indexes
Views
Synonyms
Sequences
Database Links
Directory Objects
Reorganize Objects

Click the links in the Database Objects heading to see the objects in your database. You can also access any schema object from the Search section of any database object home page.

When you select a schema link, a standard *Object_Name* page appears. As shown in Figure 8–2, you can search for objects by entering values in the **Schema** and **Object Name** boxes.

Figure 8–2 Search Section for Database Object Pages



Object Type: Table

Search

Select an object type and optionally enter a schema name and an object name to filter the data that is displayed in your results set.

Schema: SYS

Object Name:

You can also access other schema objects, including those with no direct link under the Database Objects heading. To select a specific database object, select an option from the **Object Type** list.

Naming Schema Objects

Every object in the database belongs to one schema and has a unique name within that schema. When you create an object, you must ensure it is in the schema where you intend to store it. Generally, you place all of the objects that belong to a single application in the same schema.

A database object name must abide by certain rules, including that it be unique within its schema. In addition, object names cannot be longer than 30 bytes and must begin with a letter. If you attempt to create an object with a name that violates any of these rules, then Oracle raises an error.

> **See Also:** *Oracle Database SQL Reference* for more information about naming database objects

Managing Tables

The following sections discuss database tables and how to create and modify them:

- About Tables
- Viewing Tables
- Viewing Table Data
- Creating Tables

- Modifying Table Attributes
- Loading Data Into Tables
- Dropping a Table

About Tables

Tables are the basic unit of data storage in an Oracle database. They hold all user-accessible data. Each table has columns and rows. In an employees table, for example, there can be columns called last_name and employee_id. Each row in the table would contain a value for employee name and number.

Types of Table

The most common type of table in an Oracle database is a relational table, structured like the employees table. Two other types of table are supported: object tables and XMLType tables. Any of the three table types can be defined as permanent or temporary. Temporary tables can be made available to multiple users or only to the user who creates the table.

You can build relational tables in either heap or index-organized structures. In the former, the rows are not stored in any particular order. In index-organized tables, the row order is determined by the values in one of more selected columns.

This section concentrates on permanent, heap-organized tables. For information about other table types and when to use them, see the *Oracle Database Administrator's Guide*, *Oracle Database Concepts*, and the *Oracle Database Performance Tuning Guide*. For the syntax required to create and alter tables, see the *Oracle Database SQL Reference*.

Column Attributes

Each column in a table is defined with one or more characteristics as follows:

Data Type

This required attribute defines the nature of the data to be stored in the column and can indicate the longest value that can be placed in the column.

Constraint

This optional attribute determines valid values for the column. Some constraints can be defined at the column level or at the table level. In Enterprise Manager, the only constraint defined at the column level on the Create Table property page is the NOT NULL constraint, which requires that a value is included in the column for every row in the table.

Default Value

This value is automatically stored in the column whenever a new row is inserted without a value being provided for the column.

These column characteristics are described in more detail in the following sections.

Data Types When you create a table, you must specify a data type for each of its columns. When you create a procedure or stored function, you must specify a datatype for each of its arguments.

These data types define the domain of values that each column can contain or each argument can have. For example, DATE columns cannot accept the value February 29 (except for a leap year) or the values 2 or SHOE. Each value subsequently placed in a column assumes the column datatype. For example, if you insert 17-JAN-2004 into a date column, then Oracle treats that character string as a date value after verifying that it translates to a valid date.

Table 8–1 lists some common Oracle built-in datatypes.

Table 8–1 Common Datatypes

Datatype	Description
VARCHAR2(size [BYTE | CHAR])	Variable-length character string having maximum length size bytes or characters. Maximum size is 4000 bytes or characters. The minimum is 1 byte or 1 character. You must specify size for VARCHAR2. BYTE indicates that the column will have byte length semantics; CHAR indicates that the column will have character semantics.
NUMBER (p,s)	Number having precision p and scale s. The precision p can range from 1 to 38. The scale s can range from -84 to 127.
DATE	Valid date range from January 1, 4712 BC to December 31, 9999 AD.
CLOB	A character large object containing single-byte or multibyte characters. Both fixed-width and variable-width character sets are supported, both using the database character set. Maximum size is (4 gigabytes - 1) * (database block size).

> **See Also:** *Oracle Database SQL Reference* for a complete list of Oracle's built-in datatypes.

In most tables, you should only need columns of NUMBER, VARCHAR2, and DATE datatypes.

When defining numeric data, you can use the precision option to set the maximum number of digits in the number, and the scale option to define how many of the digits are to the right of the decimal point. For example, a field to hold monetary values might be defined as NUMBER(12,2), providing ten digits for the primary unit of currency (dollars, pounds, marks, and so on) and two digits for the secondary unit (cents, pennies, pfennigs, and so on).

To define a VARCHAR2 field for character data, you must include the size value. Set the size to the maximum number of bytes or characters to be stored in the column. A column to hold postal codes for different countries, for example, might be restricted to 12 bytes by defining it as VARCHAR2(12).

DATE columns are automatically formatted by Oracle to include a date and time component.

NOT NULL Column Constraint The NOT NULL constraint on a column requires that the column must contain a value whenever a row is inserted or updated. Unlike other constraints described in "Table-Level Constraints" on page 8-5, which may be defined as part of the column definition or part of the table definition, the NOT NULL constraint must be defined as part of the column definition.

Use a NOT NULL constraint when the data is required for the integrity of the database. For example, if all employees must belong to a specific department, then the column that contains the department identifier should be defined with a NOT NULL constraint. On the other hand, do not define a column as NOT NULL if the data may be unknown or may not exist when rows are added or changed, for example, the second, optional line in a mailing address.

A primary key constraint automatically adds a NOT NULL constraint to the column or columns included in the primary key.

Default Values When you define a column with a default value, any new rows inserted into the table store the default value unless the row contains an alternate value for the column.

Table-Level Constraints

In an Oracle database, you can apply rules to preserve the integrity of your data. For example, in a table containing employee data, the employee name column cannot accept NULL as a value. Similarly, in this table you cannot have two employees with the same ID.

Oracle Database enables you to apply data integrity rules called constraints to tables at the column level or, with the exception of the NOT NULL constraint, at the table level. Any attempt to insert or update a row that violates a constraint results in an error and the statement is rolled back. Likewise, any attempt to apply a new constraint to a populated table also results in an error if any existing row violates the new constraint.

The types of constraints you can apply at the table level are as follows:

- **Primary Key**—Requires that a column (or combination of columns) be the unique identifier of the row. A primary key column does not allow NULL values.
- **Unique Key**—Requires that every value in a column be unique. That is, no two rows can have duplicate values in a specified column or combination of columns. The set of columns is said to be the unique key.
- **Check**—Requires that a column (or combination of columns) satisfy a condition for every row in the table. A check constraint must be a boolean expression that is evaluated using the column value about to be inserted or updated to the row.
- **Foreign**—Requires that all column values in the child table exist in the parent table. The table that includes the foreign key is called the dependent or child table. The table that is referenced is called the parent table. An example of a foreign key constraint is when the department column of the employees table must contain a department ID that exists in the parent department table.

Constraints can be created and, in most cases, modified with different statuses. The options include enabled or disabled, which determine if the constraint is checked when rows are added or modified, and deferred or immediate, which cause constraint validation to occur at the end of a transaction or at the end of a statement, respectively.

> **See Also:** *Oracle Database Concepts* for more information on constraints.

Storage Attributes

When you create a table in a locally managed tablespace and use automatic segment space management, you should allow Oracle to build the table using its own storage algorithms. If you need to store a table in a dictionary-managed tablespace, then you can set the storage characteristics for the table.

The preconfigured database you installed does not contain dictionary-managed tablespaces. Therefore, these options are not discussed here but are described in the *Oracle Database Administrator's Guide* and *Oracle Database Concepts* manual if you need to use them.

Other Table Creation Considerations

When creating a table with the Create Table page, you can click the **Show SQL** button to display the SQL statement that has been constructed. This option shows the statement even if it is incomplete, so you need to complete your input to see the SQL statement when using this option.

You can also create a new table with the same the structure and contents as an existing table. This technique can be useful to build a test table that resembles a production table. To do this, select **SQL** in the **Define Using** list and complete the statement that is begun for you in the **CREATE TABLE AS** box. Your query can include all columns and rows from the original table, or you can use the SQL `SELECT` statement to identify which rows you want, rename columns with column aliases, or exclude certain columns.

Your new table may include a column defined with an abstract data type (ADT). ADTs are created with the `CREATE TYPE` statement and are described in detail in the *Oracle Database SQL Reference*.

If you are creating a table containing one or more LOB columns, then you may define different storage options for the LOB components of the table. In most cases, you should use the default settings. If you want to override them, click **Set Default LOB Attributes** and complete the entries in the Set Default LOB Attributes page. Additionally, you can set LOB attributes for a single column by selecting the column name and clicking **Advanced Attributes**.

You may also create tables with multiple partitions. Each partition is a separate database object with many of the characteristics of an individual table, although each partition contains the same column definitions as its parent table. Partitioning a very large table can simplify the management of the table, because each partition can be configured independently of the other partitions. Also, some operations against a partitioned table may be more efficient than against the table if it were not partitioned. This is because the partitions may be spread across more disk drives than a non-partitioned table and because the optimizer may be able to avoid processing the contents of every partition when processing certain SQL statements.

For some applications, particularly data warehousing, with large tables that are frequently queried but very rarely updated, you may create compressed tables. These require less disk storage than uncompressed tables (which are the default). To create a compressed table, you will need to write your own `CREATE TABLE` statement because the option is not available in Enterprise Manager.

> **See Also:** *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for design and management considerations of different table types and *Oracle Database SQL Reference* for the syntax to create tables

Viewing Tables

You can use Enterprise Manager to view all of the tables in a specified schema. You can also view the table definitions.

To view tables:

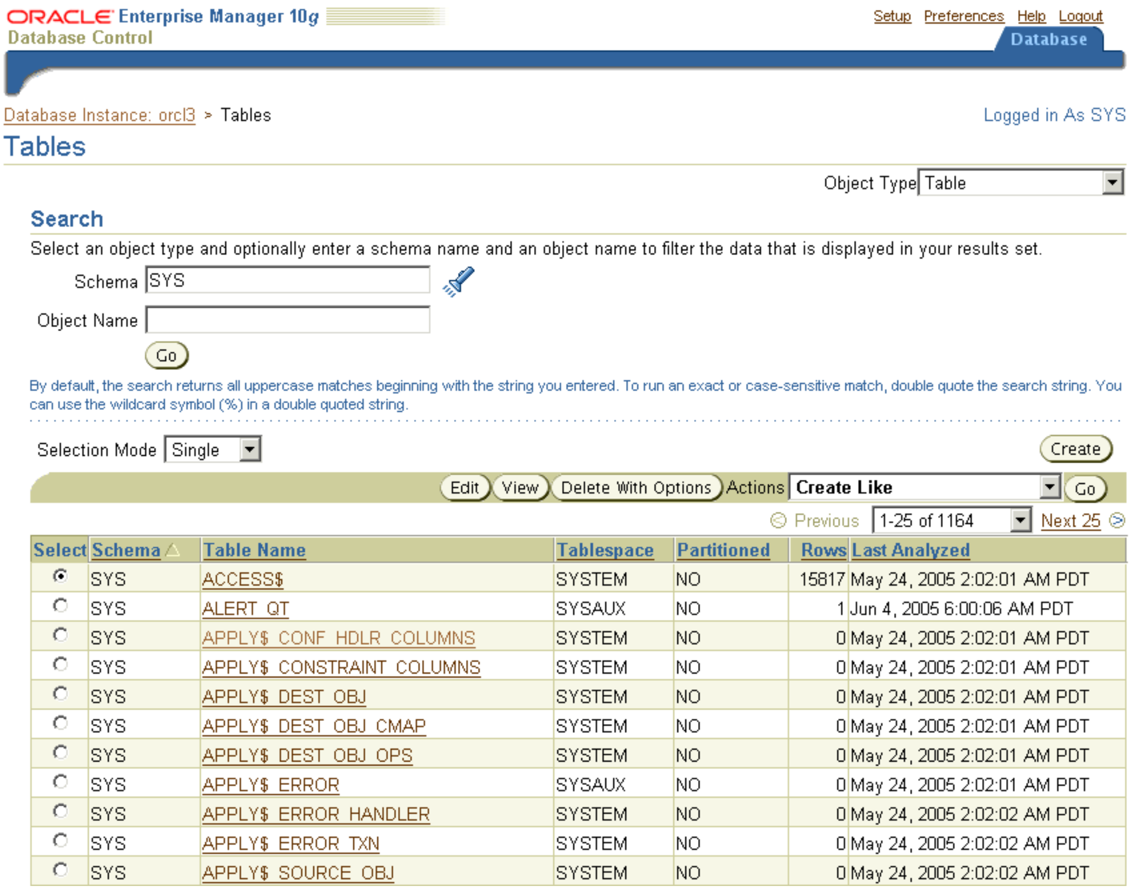
1. In the Administration page, click **Tables**.

   The Tables property page appears.

2. In the **Schema** box, enter the name of a schema. For example, enter `SYS`. Alternatively, click the flashlight icon adjacent to the **Schema** box to search for a schema.
3. Leave **Object Name** blank to search for all tables in a schema. Alternatively, enter an object name or click the flashlight icon to search for one.
4. Click **Go**.

   The tables in the specified schema are displayed. Some of the tables in the `SYS` schema are shown in Figure 8–3.

Figure 8–3 Tables Property Page



Object Type: Table

Select an object type and optionally enter a schema name and an object name to filter the data that is displayed in your results set.

Schema: SYS

Object Name:

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode: Single

Actions: Create Like

1-25 of 1164

Select	Schema △	Table Name	Tablespace	Partitioned	Rows	Last Analyzed
●	SYS	ACCESS$	SYSTEM	NO	15817	May 24, 2005 2:02:01 AM PDT
○	SYS	ALERT_QT	SYSAUX	NO	1	Jun 4, 2005 6:00:06 AM PDT
○	SYS	APPLY$_CONF_HDLR_COLUMNS	SYSTEM	NO	0	May 24, 2005 2:02:01 AM PDT
○	SYS	APPLY$_CONSTRAINT_COLUMNS	SYSTEM	NO	0	May 24, 2005 2:02:01 AM PDT
○	SYS	APPLY$_DEST_OBJ	SYSTEM	NO	0	May 24, 2005 2:02:01 AM PDT
○	SYS	APPLY$_DEST_OBJ_CMAP	SYSTEM	NO	0	May 24, 2005 2:02:01 AM PDT
○	SYS	APPLY$_DEST_OBJ_OPS	SYSTEM	NO	0	May 24, 2005 2:02:01 AM PDT
○	SYS	APPLY$_ERROR	SYSAUX	NO	0	May 24, 2005 2:02:01 AM PDT
○	SYS	APPLY$_ERROR_HANDLER	SYSTEM	NO	0	May 24, 2005 2:02:02 AM PDT
○	SYS	APPLY$_ERROR_TXN	SYSTEM	NO	0	May 24, 2005 2:02:02 AM PDT
○	SYS	APPLY$_SOURCE_OBJ	SYSTEM	NO	0	May 24, 2005 2:02:02 AM PDT

5. In the results list, select a table. Click **View** or click the link in the **Table Name** column to view a table definition.

   The View Table: *table_name* page appears.

Viewing Table Data

Besides viewing table names and table definitions, you can view the data stored in the table as well as the SQL statement used to display the data. You can also change the SQL statement to alter the result set.

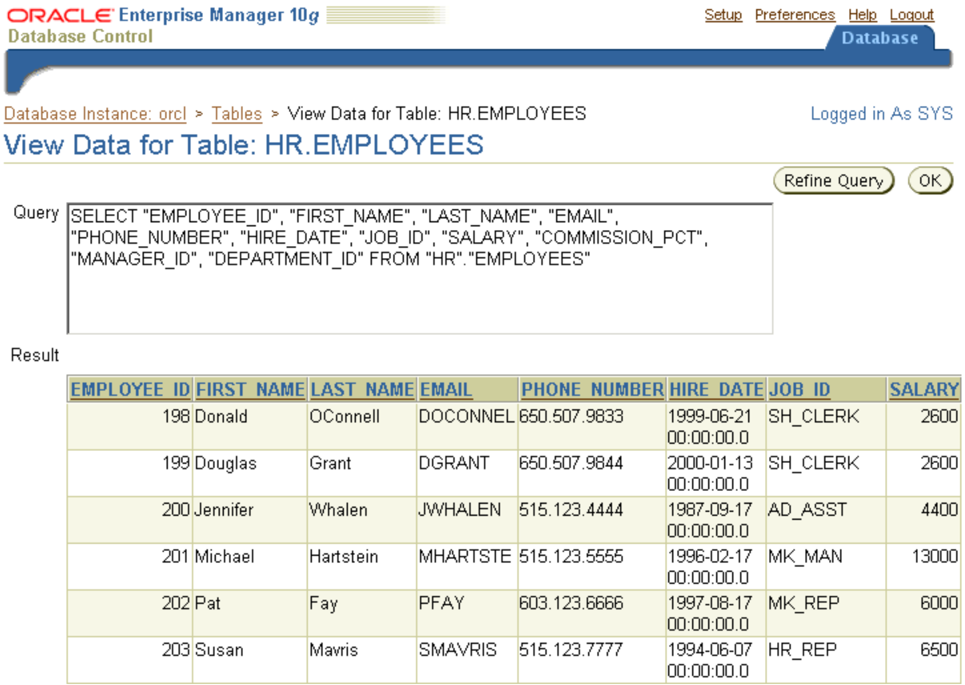
To view table data:

1. Search for a table as explained in "Viewing Tables" on page 8-6. For example, search for the tables in the `hr` schema.

2. In the results list, select the table whose data you want to view. For example, select `EMPLOYEES`.
3. In the **Actions** list, select **View Data**.
4. Click **Go**.

   The View Data for Table: *table_name* page appears. Part of the page for `hr.employees` is shown in Figure 8–4. The **Query** box displays the SQL query that was executed in order to view the data for the table. The Result section shows the data in the table.

Figure 8–4 View Data for Table Page


ORACLE Enterprise Manager 10g
Database Control
Setup
Preferences
Help
Logout
Database
Database Instance: orcl > Tables > View Data for Table: HR.EMPLOYEES
Logged in As SYS
Refine Query
OK
Query
Result

View Data for Table: HR.EMPLOYEES

```
SELECT "EMPLOYEE_ID", "FIRST_NAME", "LAST_NAME", "EMAIL",
"PHONE_NUMBER", "HIRE_DATE", "JOB_ID", "SALARY", "COMMISSION_PCT",
"MANAGER_ID", "DEPARTMENT_ID" FROM "HR"."EMPLOYEES"
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
198	Donald	OConnell	DOCONNEL	650.507.9833	1999-06-21 00:00:00.0	SH_CLERK	2600
199	Douglas	Grant	DGRANT	650.507.9844	2000-01-13 00:00:00.0	SH_CLERK	2600
200	Jennifer	Whalen	JWHALEN	515.123.4444	1987-09-17 00:00:00.0	AD_ASST	4400
201	Michael	Hartstein	MHARTSTE	515.123.5555	1996-02-17 00:00:00.0	MK_MAN	13000
202	Pat	Fay	PFAY	603.123.6666	1997-08-17 00:00:00.0	MK_REP	6000
203	Susan	Mavris	SMAVRIS	515.123.7777	1994-06-07 00:00:00.0	HR_REP	6500

5. Click a column name to sort the data in a column.
6. Click **Refine Query** to change the query.

   The Refine Query for Table: *table_name* page appears. This page enables you to select the columns to display. It also enables you to specify a `WHERE` clause for the SQL statement to limit the results.

You can also write your own SQL query using a `SELECT` statement to see the contents of a table. You can execute SQL statements by starting an *i*SQL*Plus session by clicking **iSQL*Plus** in the Related Links section of any of the Database property pages. A detailed description of the `SELECT` statement and its options is contained in the *Oracle Database SQL Reference*.

Creating Tables

You can use Enterprise Manager to create tables. You can do so as a database administrator or user with the `CONNECT` role.

Estimating the Size of New Tables

During table creation with Enterprise Manager, you can estimate the size of the table. This action enables you to determine whether you have sufficient room in your database or on the available disks to store the table. If you do not have room, then you can still create the table but not populate it until you have obtained the necessary storage.

To estimate the table size:

1. In the Database Objects section of the Administration page, click **Tables**.

   The Tables page appears.

2. Click **Create**.

   The Create Table: Table Organization page appears.

3. Select **Standard, Heap Organized** and click **Continue**.

   The Create Table page appears.

4. Enter the table and schema names as well as your column names and datatypes, then click **Estimate Table Size**.

   The Estimate Table Size page appears.

5. In **Projected Row Count**, enter the projected number of rows in the table and click **Estimate Table Size**

   Enterprise Manager returns its estimate in MB.

Creating a Table: Example

In the following example, you create a table called `purchase_orders` in the `MYUSER` schema that you created in Chapter 7, "Administering Users and Security". The table will have columns `po_number`, `po_description`, `po_date`, and `po_item`.

To create a table in the `MYUSER` schema:

1. In the Database Objects section of the Administration page, click **Tables**.

   The Tables page appears.

2. Click **Create**.

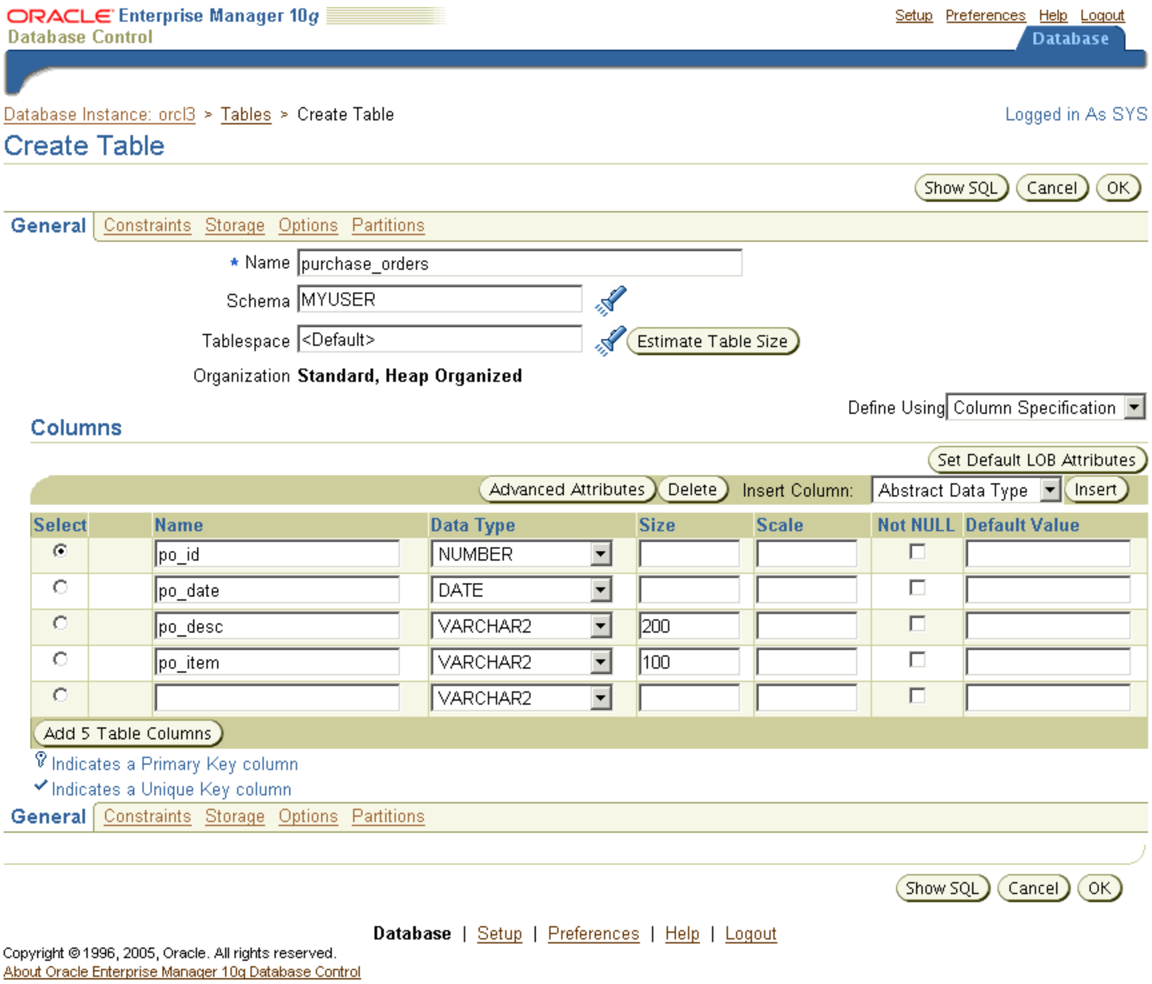
   The Create Table: Table Organization page appears.

3. Select **Standard, Heap Organized**.

4. Click **Continue**.

   The Create Table page appears. See Figure 8–5, "Create Table Page".

Figure 8–5 Create Table Page



Database Instance: orcl3 > Tables > Create Table

Logged in As SYS

Create Table

* Name: purchase_orders

Schema: MYUSER

Tablespace: <Default>

Organization: **Standard, Heap Organized**

Define Using: Column Specification

Columns

Select	Name	Data Type	Size	Scale	Not NULL	Default Value
●	po_id	NUMBER			□	
○	po_date	DATE			□	
○	po_desc	VARCHAR2	200		□	
○	po_item	VARCHAR2	100		□	
○		VARCHAR2			□	

Indicates a Primary Key column

✓ Indicates a Unique Key column

Copyright © 1996, 2005, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Database Control

5. In the **Name** box, enter `purchase_orders` as the table name. In the **Schema** box, enter `MYUSER`. In the **Tablespace** box, accept **Default**.

6. In the Columns section, enter the following columns with the datatypes:

   - `po_id` as `NUMBER`
   - `po_date` as `DATE`
   - `po_desc` as `VARCHAR2(200)`
   - `po_item` as `VARCHAR2(100)`

   You can leave **Scale**, **Not NULL**, and **Default Value** blank.

   > **Note:** If you want to the create table with partitions, you can do so at this step by clicking **Partitions**.

7. Click **OK**.

   An update message appears indicating that the table has been successfully created.

   See Also: "About Tables" on page 8-3 for more information.

Modifying Table Attributes

You can use Enterprise Manager to add and delete columns or add constraints. The following sections show how to add a column, and how to add and drop a table constraint.

Adding a Column

In this example, you modify the `purchase_orders` table by adding a new column called `po_item2`.

To add the `po_item2` column:

1. On the **Tables** page, enter `MYUSER` in the **Schema** box and click **Go**.

   The result list shows the `PURCHASE_ORDERS` table that you created previously.

2. Select the `PURCHASE_ORDERS` table and click **Edit**.

   The Edit Table: MYUSER.PURCHASE_ORDERS page appears.

3. In the first available **Name** box that is blank, enter `po_item2` as the new column name. In the **Data Type** list, select **VARCHAR2**. In the **Size** box, enter `100`. You can leave **Scale**, **Not NULL**, and **Default Value** blank.

4. Click **Apply**.

   An Update Message appears indicating that the table has been modified successfully.

See Also: "About Tables" on page 8-3 for more information.

Dropping a Column

This example shows you how to drop a column, in this case, the `po_item2` column that was added to the `PURCHASE_ORDERS` table in "Adding a Column".

To delete the `po_item2` column:

1. On the **Tables** page, enter `MYUSER` in the **Schema** box and click **Go**.

   The result list should shows the `PURCHASE_ORDERS` table that you created previously.

2. Select the `PURCHASE_ORDERS` table and click **Edit**.

   The Edit Table: MYUSER.PURCHASE_ORDERS page appears.

3. In the Columns section, select the column you want to delete. In this case, select `po_item2`.

4. Click **Delete**.

   The row that contained the information for the deleted column, `po_item2`, becomes blank to indicate that the table has been successfully modified.

Creating a New Table Constraint

You can add constraints to a table with Enterprise Manager. In the current example, assume that you want to enforce the rules that purchase orders are always entered with an ID and that the purchase order ID (`po_id`) is the primary key of the table. You will add the `NOT NULL` and primary key constraints to this column.

Note: You can also add constraints during table creation.

To add these constraints to the `purchase_orders` table:

1. On the **Tables** page, enter `MYUSER` in the **Schema** box and click **Go**.

   The result list shows the `PURCHASE_ORDERS` table that you created previously.

2. Select the `PURCHASE_ORDERS` table and click **Edit**.

   The Edit Table: MYUSER.PURCHASE_ORDERS page appears.

3. Select the `po_id` column and check **Not NULL** to apply the `NOT NULL` constraint to the column.

4. Click **Apply**

   A confirmation message appears.

5. Click **Constraints**.

   The Constraints property page appears.

6. Select **PRIMARY** from the drop-down menu and click **Add**. This action applies the primary key constraint.

   The Add Primary Constraint page appears.

7. In the **Available Columns** list, select **PO_ID** and move it to the **Selected Columns** list. To keep the default settings for the constraints, do not change the check boxes in the Attributes area of the page.

8. Click **Continue**

   The new constraint appears on the Constraints page.

9. Click **Apply**.

   A confirmation message appears. You can view the new constraints by navigating to the View Table page.

> **See Also:** "Table-Level Constraints" on page 8-5 for more information.

Modifying an Existing Constraint

You can change the status of an existing constraint, for example, from an enabled to a disabled state. This example uses the primary key constraint created in the previous section, "Creating a New Table Constraint".

To disable a constraint:

1. On the **Tables** page, enter `MYUSER` in the **Schema** box and click **Go**.

   The result list should shows the `PURCHASE_ORDERS` table that you created previously.

2. Select the `PURCHASE_ORDERS` table and click **Edit**.

   The Edit Table: MYUSER.PURCHASE_ORDERS page appears.

3. Click **Constraints**.

   The Constraints property page appears.

4. Select the row containing the `PRIMARY` constraint from the Constraints table and click **Edit**.

   The Edit PRIMARY Constraint page appears.

5. In the Attributes section, check **Disabled**.
6. Click **Continue**.
7. Click **Apply**.

   A confirmation message appears. The Disabled column shows that the primary key constraint has been disabled.

Dropping a Constraint

You can drop constraints from a table with Enterprise Manager. Although you do not have to disable a constraint before dropping it, you can determine whether the constraint can be dropped by attempting to disable it first. If a constraint in a parent table enforces a foreign key constraint in a child table, and if the child table contains dependent rows, then the constraint cannot always be disabled or dropped.

Continuing the current example, you drop the primary key constraint that you created in the section, "Creating a New Table Constraint".

To drop a constraint:

1. On the **Tables** page, enter `MYUSER` in the **Schema** box and click **Go**.

   The result list should shows the `PURCHASE_ORDERS` table that you created previously.
2. Select the `PURCHASE_ORDERS` table and click **Edit**.

   The Edit Table: MYUSER.PURCHASE_ORDERS page appears.
3. Click **Constraints**.

   The Constraints property page appears.
4. Select the row containing the `PRIMARY` constraint from the Constraints table and click **Delete**.
5. Click **Apply**.

   A confirmation message appears.

Loading Data Into Tables

You can use Enterprise Manager to load data into tables in batch. Batch loading is useful when you have a lot of data. You can load data from the file system. You can also export data into files.

For loading data with Enterprise Manager, you can use a comma-delimited data file. Enterprise Manager can also create your control file automatically. This SQL*Loader control file tells Oracle how the data is to be loaded. It is different from a database control file.

The following example loads data into the `purchase_orders` table we created with `MYUSER` in "Creating a Table: Example" on page 8-9. For simplicity, this example loads only three rows. Your control and data files can be different, but the idea is the same for this method of loading.

This example requires you to create a data file named `load4.dat` on the file system of the database host or on your client host. The contents of the file should be as follows:

```
1, 25-MAY-2005, Office Equipment, Optical Mouse
2, 18-JUN-2005, Computer System, Dell Laptop
3, 26-JUN-2005, Travel Expense, Car Rental
```

To load the data file into the `myuser.purchase_orders` table:

1. From the Move Row Data section of the Maintenance page, click **Load Data from User Files**.

   The Load Data: Generate Or Use Existing Control File page appears.

2. Select **Automatically Generate Control File**. Also enter the Host Credentials (username and password) for the host machine. Click **Continue**.

   The Load Data: Data Files page appears.

3. Select **Data File is Located on Database Server Machine** and enter the path to your data file. Alternatively, if you created the data file on your client host, select **Data File is Located on Your Browser Machine** and enter the file path. Click **Next**.

   The Load Data: Table and File Format page appears. This page displays the contents of your data file.

4. In **Table Name**, specify the destination table for your data. Enter `myuser.purchase_orders` and click **Next**.

   The Load Data: File Format Attributes page appears.

5. Accept all defaults. In this example, we use the comma character as the delimiter. Click **Next**.

   The Load Data: Load Method page appears.

6. Accept the default **Conventional Path** as the loading method. If you have a lot of data, you can select **Direct Path**. Click **Next**.

   The Load Data: Options page appears.

7. In the Optional Files section, leave **Generate log file where logging information is to be stored** checked. You can accept the default file name and path or enter a different one. Note that this page gives you the option of limiting the number of rows loaded. Because we are only loading three rows in this example, we leave this box unchecked. Click **Next**.

   The Load Data: Schedule page appears.

8. Enter values for **Job Name** and **Description**. Select **Immediately** to run the job now. Click **Next**.

   The Load Data: Review page appears. This page lets you review your file names and loading methods. You can also view your loading control file. If you want to change something, you can click the **Back** button.

9. Click **Submit Job** to start the loading.

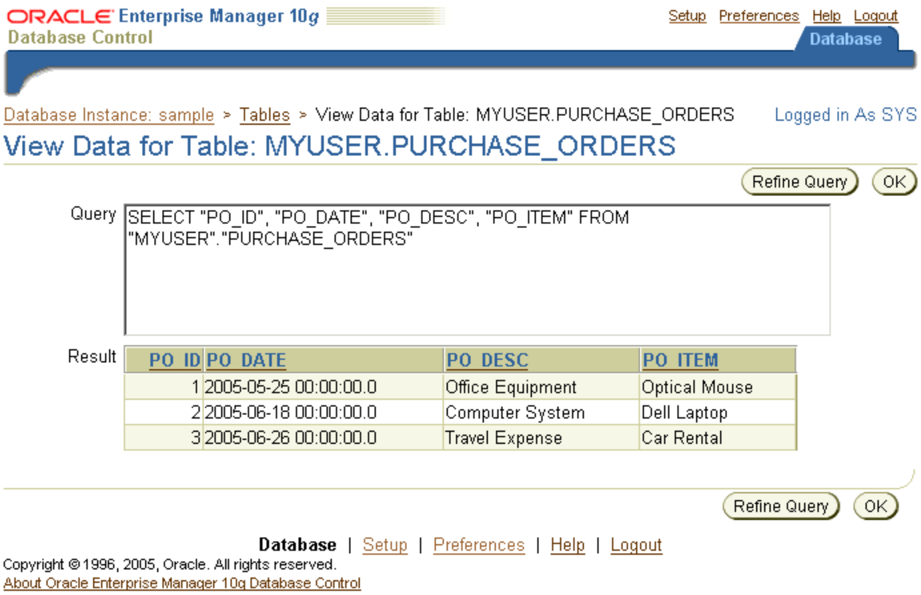
   The Job Activity page appears, confirming that your job was created successfully.

10. Search for your job name to view its status.

    The Job Activity page should display your job information and show that your job succeeded. You can get more information by clicking the link under **Status**.

11. You can confirm the loaded data by navigating to the Tables page, searching for the `myuser.purchase_orders` table, and choosing **View Data** as the action. See Figure 8–6, "View Data: MYUSER.PURCHASE_ORDERS".

Figure 8–6 View Data: MYUSER.PURCHASE_ORDERS


ORACLE Enterprise Manager 10g
Database Control
Setup Preferences Help Logout
Database
Refine Query
OK
Database | Setup | Preferences | Help | Logout

Database Instance: sample > Tables > View Data for Table: MYUSER.PURCHASE_ORDERS Logged in As SYS

View Data for Table: MYUSER.PURCHASE_ORDERS

Query

```
SELECT "PO_ID", "PO_DATE", "PO_DESC", "PO_ITEM" FROM
"MYUSER"."PURCHASE_ORDERS"
```

Result

PO_ID	PO_DATE	PO_DESC	PO_ITEM
1	2005-05-25 00:00:00.0	Office Equipment	Optical Mouse
2	2005-06-18 00:00:00.0	Computer System	Dell Laptop
3	2005-06-26 00:00:00.0	Travel Expense	Car Rental

Copyright © 1996, 2005, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Database Control

See Also: "About Tables" on page 8-3 for more information.

Dropping a Table

If you no longer need a table or its contents, then you can drop the table using Enterprise Manager. Be certain that you really do not need the data in the table before you drop it. It may be difficult and time-consuming to retrieve the records, if they can be retrieved, after you execute the drop statement.

To test this procedure, follow the procedure in "Creating Tables" on page 8-8 to create a table named `test` in the `MYUSER` schema. Create the `test` table with one column called `col` of type `VARCHAR2(1)`.

To drop a table:

1. On the **Tables** page, enter `MYUSER` in the **Schema** box and click **Go**.

   The result list should show the `test` table that you just created.

2. Select the `TEST` table and click **Delete With Options**.

   The Delete With Options page appears.

3. Select **Delete the table definition, all its data, and dependent objects (DROP)**.

4. Check **Delete all referential integrity constraints (CASCADE CONTRAINTS)**.

5. Click **Yes**.

   The Edit table page displays a confirmation message when the table is successfully deleted.

Managing Indexes

The following sections describe how to create and manage indexes:

- About Indexes
- Viewing Indexes

- Creating Indexes
- Dropping Indexes

About Indexes

Indexes are optional structures associated with tables. You can create them to improve query performance. Just as the index in this book helps you to quickly locate specific information, an Oracle index provides a quick access path to table data.

You can create indexes on one or more columns of a table. After an index is created, it is automatically maintained and used by Oracle. Changes to a table's data or structure, such as adding new rows, updating rows, or deleting rows, are automatically incorporated into all relevant indexes with complete transparency to the user.

Some indexes are created implicitly through constraints that are placed on a table. For example, a column with the constraint that its values be unique causes Oracle to create a unique key index.

When you click the **Indexes** link in the Database Objects section of the Administration page, an Indexes page appears. This page is similar in content and functionality to the Tables page shown in Figure 8–3.

Indexes and Performance

Before you add additional indexes, you should examine the performance of your database. You can then compare performance after the new indexes are added.

Indexes are generally of value to queries and to SQL statements that need to operate on a single, existing row or a small number of existing rows. However, too many indexes can increase the processing overhead for statements that add, modify, or delete rows.

To determine if your application can improve performance with more indexes, you can run the SQL Access Advisor in Enterprise Manager. See Using the SQL Access Advisor in Chapter 10, "Monitoring and Tuning the Database".

Index Types

Indexes can be categorized in a number of ways. The primary options are listed in the following sections.

Standard (B-tree) and Bitmap A standard, B-tree index contains an entry for each value in the index key along with an address to the row where the value is stored. A B-tree index is the default and most common type of index in an Oracle database.

A bitmap index uses strings of bits to encapsulate values and potential row addresses. It is more compact than a B-tree and can perform some types of retrieval more efficiently. For general use, however, a bitmap index requires more overhead during row operations on the table and should be used primarily for data warehouse environments as described in the *Oracle Database Data Warehousing Guide*.

Ascending and Descending The default search through an index is from lowest to highest value, where character data is sorted by ASCII values, numeric data from smallest to largest number, and date from the earliest to the latest value. This default behavior is performed in indexes created as ascending indexes. You can cause index searches to reverse the search order by creating the related index with the descending option.

Column and Functional Typically, an index entry is based on the value or values found in the table's column or columns. This is a column index. Alternatively, you can create a function-based index in which the indexed value is derived from the table data. For example, to find character data that can be in mixed case, you could use a function-based index to look for the values as if they were all in uppercase characters.

Single Column and Concatenated You can create an index on just one column, which is called a single column index, or on multiple columns, which is called a concatenated index. Concatenated indexes are useful when all of the columns are likely to be included in the `WHERE` clause of frequently executed SQL statements.

For concatenated indexes, you should define the columns used in the index carefully so that the column with the fewest duplicate values is named first, the column with next fewest duplicate values is second, and so on. Columns with many duplicate values or many rows with `NULL` values should not be included or should be the last-named columns in the index definition.

Non-partitioned and Partitioned As with tables, you can create your indexes with or without partitions. In most situations, it is useful to partition an index when the associated table is also partitioned and the index uses the same partitioning scheme as the table, known as a local index. You need not partition an index like the underlying table. You may even create a non-partitioned, or global, index on a partitioned table.

> **See Also:** *Oracle Database Concepts* and *Oracle Database Administrator's Guide* for design and management considerations of different index types and *Oracle Database SQL Reference* for the syntax to create tables

Viewing Indexes

To find out which indexes exist in your database, use the search options on the Indexes page.

To view indexes:

1. In the Administration page, click **Indexes**.

   The Indexes page appears.

2. In the **Search By** list, select **Index Name** to search for indexes by name.

   Alternatively, you can select **Table Name** to search for the name of the table on which the index is built.

3. In the **Schema** box, enter the name of a schema. For example, enter `SYS`. Alternatively, click the flashlight icon adjacent to the **Schema** box to search for a schema.

4. Leave **Object Name** blank to search for all indexes in a schema. Alternatively, enter an object name or click the flashlight icon to search for one.

5. Click **Go**.

   The indexes in the specified schema are displayed.

6. In the results list, select a table and then click **View** or the link in the **Index** column to view a table definition.

   The View Index: *index_name* page appears. In the General area, you can find basic information about the index, such as its status and the column or columns on which it is built. Other areas include information about the space consumed by the index and the options used in its definition.

Creating Indexes

You can create indexes with Enterprise Manager. To create an index, you specify one or more columns to be indexed and the type of index you want to create.

The following example creates a standard B-tree index on the `SUPPLIER_ID` column in the `SH.PRODUCTS` table, which is part of the sample schema.

To create an index:

1. In the Database Objects section of the Administration page, click **Tables**.

   The Tables page appears.

2. Enter `sh` in the **Schema** box and click **Go**.

   The result list shows the tables in the `sh` schema.

3. Select the `PRODUCTS` table and select **Create Index** from the **Actions** list. Click **Go**.

   The Create Index page appears.

4. Enter the following information:

   - In the **Name** box, enter `PRODUCTS_SUPPLIER_IDX`.
   - For **Tablespace**, accept the default.
   - For **Index Type**, select **Standard - B-tree**.
   - In the **Table Columns** list, select the `SUPPLIER_ID` column by entering `1` in the **Order** column.
   - For **Sorting Order**, accept the default selection of **ASC**.

5. Click **OK** to create the index.

   The Indexes page appears with a confirmation message. The new index is listed under Results.

See Also: "About Indexes" on page 8-16 for more information.

Dropping Indexes

If you no longer need an index, you can drop it by using Enterprise Manager.

In this example, you drop the `PRODUCTS_SUPPLIER_IDX` index that you created previously.

To drop an index:

1. In the Administration page, click **Indexes**.

   The Indexes page appears.

2. In the **Search By** list, select **Index Name** to search for indexes by name. In **Schema**, enter `sh`. In **Object Name**, enter `PRODUCTS_SUPPLIER_IDX`.

3. Click **Go**.

   The results page should display the `PRODUCTS_SUPPLIER_IDX` index.

4. Select the `PRODUCTS_SUPPLIER_IDX` index and click **Delete**.

   A Confirmation page appears.

5. Click **Yes** to delete the index.

The Edit table page displays a confirmation message if the index is successfully deleted.

> **Note:** You cannot drop an index that is currently used to enforce a constraint. You must disable or drop the constraint and then, if the index is not dropped as a result of that action, drop the index.

Managing Views

The following sections discuss views:

- About Views
- Displaying Views
- Creating a View: Example
- Dropping Views

About Views

Views are customized presentations of data in one or more tables or other views. You can think of them as stored queries. Views do not actually contain data, but instead derive their data from the tables upon which they are based. These tables are referred to as the base tables of the view.

Like tables, views can be queried, updated, inserted into, and deleted from, with some restrictions. All operations performed on a view actually affect the base tables of the view. Views provide an additional level of security by restricting access to a predetermined set of rows and columns of a table. They also hide data complexity and store complex queries.

Many important views are in the SYS schema. The data dictionary views are called static because they change only when a change is made to the data dictionary, for example, when a new table is created or a user is granted new privileges.

Many data dictionary tables have three corresponding views:

- A DBA_ view displays all relevant information in the entire database. DBA_ views are intended only for administrators.
- An ALL_ view displays all the information accessible to the current user, including information from the current user's schema as well as information from objects in other schemas, if the current user has access to those objects through privileges or roles.
- A USER_ view displays all the information from the schema of the current user. No special privileges are required to query these views.

The columns of the ALL_, DBA_, and USER_ views corresponding to a single data dictionary table are usually nearly identical.

The views that monitor ongoing database activity are called dynamic performance views. The names of dynamic performance views, available only to administrators, start with the characters V$ and are often referred to as the V$ views.

Complete descriptions of the views in the SYS schema are contained in the *Oracle Database Reference*.

Displaying Views

You can use Enterprise Manager to display the views in a specified schema. You can also display the view definitions.

To display views:

1. In the Administration page, click **Views**.

   The Views property page appears.

2. In the **Schema** box, enter the name of a schema. For example, enter SYS. Alternatively, click the flashlight icon adjacent to the **Schema** box to search for a schema.

3. Leave **Object Name** blank to search for all views in a schema. Alternatively, enter an object name or click the flashlight icon to search for one.

4. Click **Go**.

   The views in the SYS schema are displayed. These views enable you to look at data dictionary information or ongoing database activity. The views in the SYS schema are also the source of information for many of the Oracle Enterprise Manager pages.

5. In the results list, select a view and then click **View** or click the link in the **View Name** column to display a view definition.

   The View: *view_name* page appears.

Creating a View: Example

The following example creates a view derived from the hr.employees table, which is part of the sample schema. The view filters all direct reports for the manager King, whose employee_id is 100. In an application scenario, this view adds an additional level of security to the hr.employees table while providing a suitable presentation of relevant information for manager King.

To create a view:

1. In the Administration page, click **Views**.

   The Views property page appears.

2. Click **Create**.

   The Create View page appears.

3. Enter the following information:

   - In **Name**, enter KING_VIEW.
   - In **Schema**, enter HR.
   - In **Query Text**, enter the following SQL statement:

     ```
     SELECT employee_id, first_name, last_name, email, phone_number, hire_date,
            job_id, salary, commission_pct, manager_id, department_id
     FROM   hr.employees
     WHERE  manager_id = 100
     ```

4. Click **OK**.

   The Views page appears with a confirmation message. The new view is listed in the results section.

5. Select KING_VIEW and then select **View Data** from the **Actions** list.

6. Click **Go**.

   The View Data for View: HR.KING_VIEW page appears. The selected data in the view appears in the Result section.

Dropping Views

If you no longer need a view, then you can drop it using Enterprise Manager.

In this example, you drop the `HR.SKING_VIEW` view that you created previously.

To drop a view:

1. In the Administration page, click **Views**.

   The Views page appears.

2. In the **Schema** box, enter `hr`. In **Object Name**, enter `SKING_VIEW`.
3. Click **Go**.

   The results page should display the `SKING_VIEW` view.

4. Select `SKING_VIEW` and click **Delete**.

   A Confirmation page appears.

5. Click **Yes** to delete the view.

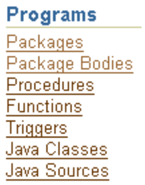
   The Views page displays a confirmation message if the view is successfully deleted.

Managing Database-Resident Program Units

Oracle offers the capability to store programs in the database. This functionality enables commonly required code to be written and tested once and then accessed by any application that requires the code. Database-resident program units also ensure that the same processing is applied to the data when the code is invoked, making the development of applications easier and providing consistency between developers.

You can write database-resident programs in PL/SQL or Java. You can use Enterprise Manager to manage source types such as PL/SQL packages, procedures, triggers, functions, and Java sources and classes. The actions include creating, compiling, creating synonyms for, granting privileges on, and showing dependencies for these source types. You can perform these actions by clicking the links in the Programs section of the Administration page, as shown in Figure 8–7.

Figure 8–7 Programs Section



This book describes the main types of PL/SQL program units. The following sections cover:

- Managing Packages
- Managing Package Bodies

- Managing Standalone Subprograms
- Managing Triggers

Note that creating and managing program units is the primary responsibility of an application developer. However, as a DBA you might be called upon to assist in managing these objects, since they can become invalidated through schema changes.

> **See Also:** *Oracle Database PL/SQL User's Guide and Reference* to learn about PL/SQL code and program units.

Managing Packages

A package is a structure that contains definitions, blocks of PL/SQL code, or both in a single unit. The contents of the package are invoked by naming the package and the element within it by means of a dot notation. For example, to execute the procedure called `STORE_SQE` in the `CTX_QUERY` package owned by the `CTXSYS` user, you would invoke it with the statement `CTXSYS.CTX_QUERY.STORE_SQE(`*`arg1`*`,`*`arg2`*`)` where *`arg1`* and *`arg2`* are the values for the two procedure arguments. To hide the complexity of packaged element names, you might want to use synonyms, described later in "Working with Other Schema Objects" on page 8-26.

Viewing Packages

To see which packages exist in your database, use the search options on the Packages page.

To view packages:

1. In the Programs sections of the Administration page, click **Packages**.

   The Packages page appears.

2. In the **Schema** box, enter the name of a schema. For example, enter `SYS`. Alternatively, click the flashlight icon adjacent to the **Schema** box to search for a schema.

3. Leave **Object Name** blank to search for all packages in the specified schema. Alternatively, enter an object name or click the flashlight icon to search for one.

4. Click **Go**.

   The packages in the specified schema are displayed.

5. In the results list, select a package and then click **View** or the link in the **Package Name** column to view a package definition. For example, click the link for `DBMS_ALERT`.

   The View Package: *package_name* page appears. This page contains the package name, schema, status, methods, and source. The source contains the code for a package and may contain the following elements:

   - Authentication information—Determines which security schema is used when the package contents are executed
   - Procedure definitions—Specify the name and list of arguments for one or more procedures.
   - Function definitions—Specify the name, list of arguments, and return data type for one or more functions.
   - Variable definitions—Specify variables that are globally available to the user of the package.

- Other public elements—Define such structures as types, constants, and exceptions that are needed by package users.

Creating Packages

As with the other database objects, you can use Enterprise Manager to create packages. This section explains how to create a package definition.

To create a package definition:

1. In the Programs sections of the Administration page, click **Packages**.

   The Packages page appears.

2. Click **Create**.

   The Create Package page appears.

3. In the **Name**, **Schema**, and **Source** boxes, enter the desired data. For example, enter the following information:

   - In **Name**, enter `test_package`.
   - In **Schema**, enter `hr`.
   - In **Source**, enter a package definition. For more information on package definitions, see *Oracle Database PL/SQL User's Guide and Reference*.

4. Click **OK**.

   An Update Message confirms the creation of the package definition.

Editing Packages

As with the other database objects, you can use Enterprise Manager to edit packages. This example edits the `test_package` package that you created in the previous section.

To edit packages:

1. In the Programs sections of the Administration page, click **Packages**.

   The Packages page appears.

2. Enter search criteria in the **Schema** and **Object Name** boxes and click **Go**. For example, search for packages in the `hr` schema.

   The packages in the specified schema are displayed.

3. In the results list, select a package and then click **Edit**. For example, select `TEST_PACKAGE`.

   The Edit Package: *package_name* page appears. You can perform the following actions:

   - Modify the package definition by typing new contents in the **Methods** and **Source** box and clicking **Apply**.
   - Compile the package by clicking **Compile**, an action that may be required if the package status has become invalid for some reason.
   - Modify the package body (see next section) by clicking **Modify Package Body**.

   Note: You can also navigate to the Edit Package property page from the Edit Package Body property page, described in the next section, by clicking **Edit Package**.

Dropping Packages

Under exceptional circumstances, such as when advised by Oracle support or an application developer, you might be required to drop a package. As with the other database objects, you can use Enterprise Manager to do so. This example drops the `test_package` package.

To drop packages:

1. In the Programs sections of the Administration page, click **Packages**.

   The Packages page appears.

2. Enter search criteria in the **Schema** and **Object Name** boxes and click **Go**. For example, search for packages in the `hr` schema.

   The packages in the specified schema are displayed.

3. In the results list, select a package and then click **Delete**. For example, select `TEST_PACKAGE`.

   A Confirmation page appears.

4. Click **Yes** to delete the package

   The Update Message confirms the deletion.

Managing Package Bodies

Package bodies contain the PL/SQL code for the procedures and functions defined in the definitions of the packages to which they belong. The same navigation paths and options for adding (creating), editing (modifying), compiling, and deleting package bodies are available in Enterprise Manager as for packages, as described in the previous section. These paths enable direct navigation between the **Edit Package Body** and the **Edit Package** property pages as mentioned in the previous section.

In some cases, the developer of the package may have wrapped one or more of the individual procedures or functions. Wrapping converts the code into non-readable strings which makes it difficult for other developers to misuse the application or for competitors to see the algorithms. Wrapped code cannot be edited. The Source box in the **Edit Package Body** page will contain a message such as `Source code could not be displayed, because it is wrapped` and will not be available for modification.

Viewing Package Bodies

To find out which package bodies exist in your database, use the search options on the Package Bodies page.

To view package bodies:

1. In the Programs sections of the Administration page, click **Package Bodies**.

   The Packages page appears.

2. In the **Schema** box, enter the name of a schema. For example, enter `SYS`.

   Alternatively, click the flashlight icon adjacent to the **Schema** box to search for a schema.

3. Leave **Object Name** blank to search for all packages in the specified schema.

   Alternatively, enter an object name or click the flashlight icon to search for one.

4. Click **Go**.

The packages in the specified schema are displayed.

5. In the results list, select a package and then click **View** or the link in the **Package Body Name** column to view a package body. For example, click the link for `CONNECTIONINTERFACE`.

   The View Package Body: *package_name* page appears. This page contains the package name, schema, status, and source. The source contains the code for a package. Note that the **Source** box for the `CONNECTIONINTERFACE` package indicates that the package is `wrapped`.

Managing Standalone Subprograms

Procedures and functions that are created outside of a package are called standalone subprograms. To execute a standalone subprogram, you only need to include its schema name and object name; there is no package name. Even so, you may want to create synonyms for subprograms that are used regularly.

A PL/SQL procedure is a subprogram that performs a specific action. You specify the name of the procedure, its parameters, its local variables, and the `BEGIN-END` block that contains its code and handles any exceptions. A function is a subprogram that computes a value. Functions and procedures are structured alike, except that functions return a value.

You can use Enterprise Manager to create, modify, compile, and delete standalone subprograms just like packaged subprograms except that they do not have a parent package (and, consequently, no navigation path in Enterprise Manager from or to the package property pages). In the Programs section of the Administration page, click the links for **Procedures** and **Functions** to navigate to the relevant property pages.

Managing Triggers

A database trigger is a stored subprogram associated with a database table, view, or event. For instance, you can have Oracle fire a trigger automatically before or after an `INSERT`, `UPDATE`, or `DELETE` statement that affects a table.

> **See Also:** *Oracle Database Application Developer's Guide - Fundamentals* and *Oracle Database PL/SQL User's Guide and Reference* for more information about the uses for and creation of triggers

You can manage triggers through Enterprise Manager. They have their own property page which you can open by clicking **Triggers** in the Programs section of the Administration page. From the Triggers property page, you can create new triggers or view, edit, and delete existing triggers.

The **Create Triggers** and **Edit Triggers** property pages look very similar to the corresponding pages for packages, package bodies, and standalone subprograms. Note the following differences:

- The **Replace If Exists** checkbox enables you to replace an existing trigger; the Enable trigger enables you to immediately enable a trigger after it is created or edited.
- The box containing the PL/SQL code is labeled **Trigger Body** instead of **Source**.

The creation, editing, and compilation of triggers is similar to that of database PL/SQL units. Because triggers run automatically as a result of the triggering DML on the table, there is no way to run them directly.

Working with Other Schema Objects

You can manage other schema objects with Enterprise Manager including the following:

- Synonyms

  A synonym is an alias for any schema object such as a table or view. Synonyms provide an easy way to hide the underlying database structure from an application or a user. For example, you can create a synonym named personnel as an alias for the employees table in the hr schema. For example, assume that you create the following query:

  ```
  SELECT   employee_id, salary
  FROM     hr.employees
  ORDER BY salary
  ```

  With the synonym in place, you can rewrite the query more simply as:

  ```
  SELECT   employee_id, salary
  FROM     personnel
  ORDER BY salary
  ```

  An additional benefit of synonyms is that you can use the same synonym in a development database as in the production database, even if the schema names are different. This technique allows the application code to run unmodified in both environments. For example, the second version of the preceding query would work in a development database where the employees table was in the dev1 schema if the personnel synonym were defined.

  Because a synonym is simply an alias, it requires no storage other than its definition in the data dictionary.

- Sequences

  A sequence is a database object from which multiple users can generate unique integers. Using a sequence generator to provide the value for a primary key in a table is an easy way to guarantee that the key value is unique.

- Database links

  A database link is a schema object that instructs Oracle to connect to a remote database to access an object there. Database links are used in distributed database environments, which are described in the *Oracle Database Administrator's Guide*.

Schemas: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter and includes annotated screen shots.

To view the Schemas OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/schema/schema.htm
```

9

Performing Backup and Recovery

This chapter introduces you to Oracle Database backup and recovery using Oracle Enterprise Manager. The goals are to introduce the basic concepts of Oracle backup and recovery, show how to configure your database for backup and recovery using the Oracle-suggested disk-based backup strategy, take some other basic backups and perform maintenance of your backups, and walk you through recovery of a full database backup.

> **Note:** The instructions in "Configuring Your Database for Basic Backup and Recovery" on page 9-4 describe how to set up your database to take advantage of Oracle-suggested backup strategies. If you chose to configure automated backups when creating the database using the Database Configuration Assistant, then you do not need to perform the steps in this section. See "Using DBCA to Create and Configure a Database" on page 2-6 for more details on creating a database already configured for automated daily backups using the Oracle-suggested backup strategy.

This chapter includes the following topics:

- Overview of Database Backup and Recovery
- Configuring Your Database for Basic Backup and Recovery
- Backing Up Your Database
- Performing Restore and Recovery Operations
- Managing Your Backups
- Displaying Backup Reports
- Backup and Recovery: Oracle by Example Series

Overview of Database Backup and Recovery

The focus in Oracle backup and recovery is generally on the physical backup of database files, which permit the full reconstruction of your database. The files protected by the backup and recovery facilities built into Enterprise Manager include datafiles, control files, server parameter files (SPFILEs), and archived redo log files. With these files your database can be reconstructed. The backup mechanisms that work at the physical level protect against damage at the file level, such as the accidental deletion of a datafile or the failure of a disk drive.

Logical-level backups, such as exporting database objects like tables or tablespaces, are a useful supplement to physical backups for some purposes. Nevertheless, logical backups cannot protect your entire database. An effective backup strategy must be based on physical-level backups.

Oracle Database's flashback features provide a range of physical and logical data recovery tools as efficient, easy-to-use alternatives to physical and logical backups. The flashback features enable you to reverse the effects of unwanted database changes without restoring datafiles from backup or performing media recovery.

This chapter introduces the following logical-level flashback features:

- Oracle Flashback Table, which lets you revert a table to its contents at a time in the recent past
- Oracle Flashback Drop, which lets you rescue dropped database tables

Neither of the preceding features requires advance preparation such as creating logical-level exports to allow for retrieval of your lost data. Both can be used while your database is available. *Oracle Database Backup and Recovery Basics* discusses the flashback features of the Oracle database at greater length.

The Oracle Enterprise Manager physical backup and recovery features are built on the Recovery Manager (RMAN) command-line client. Enterprise Manager makes available much of the functionality of RMAN, and provides wizards and automatic strategies to simplify and further automate RMAN-based backup and recovery.

> **See Also:** *Oracle Database Backup and Recovery Basics* and *Oracle Database Backup and Recovery Advanced User's Guide* for more details on the full range of Oracle Database backup capabilities

Oracle Backup, Restore, and Recovery Concepts

To back up your database is to make copies of your datafiles, control file, and archived redo logs (if your database runs in `ARCHIVELOG` mode). Restoring a database from a backup means copying the physical files that make up the database from a backup medium (disk or tape) to their locations during normal database operation. Recovery of your database is the process of updating database files restored from a backup with the changes made to the database since the backup, typically using redo log files.

Consistent and Inconsistent Backups

A backup can be consistent or inconsistent. A backup is consistent when there are no changes in the redo log that have not already been applied to the datafiles at the time of the backup.

To make a consistent backup, your database must have been shut down normally. It cannot be reopened for the duration of the backup. In shutting down the database, all committed changes in the redo log are written to the datafiles, so the datafiles are in a transaction-consistent state. This process is known as an offline backup because the entire database is offline for the duration of the backup.

In contrast to a consistent backup, an inconsistent backup is made while the database is open. In an inconsistent backup, the online redo logs contain changes that have not yet been applied to the datafiles. The datafiles are not in a transaction-consistent state. The database must be run in `ARCHIVELOG` mode to preserve the redo log. The online redo log at the moment of the backup must be archived and backed up along with the datafiles to preserve these changes.

In spite of the name, an inconsistent backup is as robust a form of backup as a consistent backup. The great advantage to making inconsistent backups is that you can back up your database while the database is open for updates.

Restoring from Consistent and Inconsistent Backups When you restore your datafiles from a consistent backup, you can open the database immediately. When datafiles are restored from an inconsistent backup, you cannot open the database until committed changes recorded in the redo logs are applied to the datafiles, bringing them to a transaction-consistent state. The process of applying changes from the redo log to the datafiles restored from an inconsistent backup is called media recovery.

Media Recovery

If you restore the archived redo logs and the datafiles from backups, then you must perform media recovery before you can open the database. Any database transactions in the archived redo logs not already reflected in the datafiles are applied to the datafiles, bringing them to a transaction-consistent state before the database is opened.

Media recovery requires a control file, datafiles (typically restored from backup), and online and archived redo logs containing changes since the time the datafiles were backed up. Media recovery is most often used to recover from a media failure, such as the loss of a file or disk, or a user error, such as the deletion of the contents of a table.

There are two forms of media recovery: complete recovery and point-in-time recovery. In complete recovery, datafiles are restored from backup; all changes from the archived and online redo logs are applied to the datafiles. The database is returned to its state at the time of failure and can be opened with no loss of committed changes.

In point-in-time recovery, you return your database to its contents at a target time of your choosing in the past. You start with a backup of datafiles created prior to the target time and a complete set of archived redo log files from the time of that backup through the target time. During recovery, each change between the backup time and the target time is applied to the datafiles.

Point-in-time recovery can return your whole database to its state at any time between the time of your backup and the most recent change in the archived redo logs. All changes after the target time are discarded. Point-in-time recovery is also sometimes called incomplete recovery because you do not recover the complete set of changes to your database.

Enterprise Manager provides a convenient interface to both complete and point-in-time recovery in the form of a Recovery Wizard. However, this book focuses on complete recovery. Point-in-time recovery is discussed at more length in *Oracle Database Backup and Recovery Basics*.

The Flash Recovery Area

To simplify the management of backup and recovery related files, Oracle enables you to create a flash recovery area for your database. You must designate the following:

- A location, which is typically a directory on disk
- A maximum disk quota for the flash recovery area
- A retention policy to specify your database recoverability goals

You can then direct backup-related activities, including archiving of redo logs, to store generated files in the flash recovery area. Oracle Database automatically manages this storage, deleting files that are no longer required on disk to meet your recoverability objectives if space is required for other files. A backup moved to tape is eligible for automatic deletion from the flash recovery area. Periodically copying backups to tape frees space in the flash recovery area for other files.

A flash recovery area simplifies backup storage management tasks. Therefore, it is strongly recommended. Except as noted, examples in this chapter assume the use of a flash recovery area.

The RMAN Repository

RMAN maintains a record of all database files, backup and recovery files, and backup history for your database. This record is referred to as the RMAN repository.

Every backup action that you perform through RMAN or Enterprise Manager is recorded in the repository, along with the location of and other metadata about every backup created by RMAN on disk or tape. If you back up a file without using RMAN, such as by copying the file on disk at the host operating system level, then you can add information about that copy to the RMAN repository as well. At recovery time, you can issue a command such as `RESTORE DATABASE`. Oracle uses the records in the repository to select backups on disk and tape needed to complete the recovery.

The primary location for the RMAN repository is in the database control file. This is one more reason why protecting your control file is a vital part of your backup strategy. In some installations, a second copy of the RMAN repository for one or more Oracle databases is stored in a set of tables called the recovery catalog. The recovery catalog is located in a separate Oracle database. Use of a recovery catalog is optional and is beyond the scope of this book.

Configuring Your Database for Basic Backup and Recovery

To take maximum advantage of Oracle Database features that automatically manage backup and recovery files and processes, configure your database as follows:

- Use a flash recovery area, which automates storage management for most backup-related files.
- Run your database in `ARCHIVELOG` mode, so that you can perform online backups and have data recovery options such as complete and point-in-time media recovery.
- Use the flash recovery area as an archived log destination for your database.

You must also set a number of policies governing which files are backed up, what format is used to store backups on disk, and when files become eligible for deletion from the flash recovery area.

Planning Space Usage and Location for the Flash Recovery Area

You should place the flash recovery area on a separate disk from the working set of database files. Otherwise, the disk becomes a single point of failure for your database.

The amount of disk space to allocate for the flash recovery area depends upon the size and activity levels of your database, which determine the size of your datafiles and redo logs files as well as your recovery objectives. Your objectives dictate what kinds of backups you use, when you make them, and how long you must keep them.

Retention Policy and the Flash Recovery Area

Space management in the flash recovery area is governed by a backup retention policy. A retention policy determines when files are obsolete, meaning that they are no longer needed to meet your data recovery objectives. Retention policies can be based on redundancy of backups or on a recovery window.

Under a redundancy-based policy, the flash recovery area considers a backup of a file obsolete only when the RMAN repository has records of a specified number of more recent backups of that file. For example, assume your policy requires that two backups of each file be kept. You make backups nightly starting on a Monday night. After the Wednesday night backup succeeds, the Monday night backup becomes redundant because the Tuesday and Wednesday backups are available.

Under a recovery window-based policy, you specify a time interval measured in days. Files become obsolete only when they are no longer needed for successful complete recovery or point-in-time recovery to any point within that number of days into the past. For example, assume that you specify a recovery window of three days. A backup of all datafiles from at least three days ago must be retained, along with a full set of archived redo logs generated since that backup.

Note: With a badly designed backup strategy, a recovery window-based retention policy can require the retention of large quantities of data. For example, assume that you specify a retention policy with a three-day recovery window and make a full database backup only on the first day of each month and no other backups. By the 28th day, recovery to any point in the three day window requires the full database backup from the first day as well as 28 days of archived redo logs.

Furthermore, recovery time may be quite long after you restore from this backup. The reason is that a minimum of 25 days' worth of transactions must be applied to the restored backup to reach the beginning of the recovery window. Thus, this strategy both wastes disk space usage and results in poor recovery performance.

A redundancy-based retention policy makes it easier to predict space usage in the flash recovery area, but it does not allow you to predict how far into the past you can recover your database. A recovery window-based policy offers better protection for your data, but can make storage requirements for backups harder to predict. As already noted, a poorly designed backup strategy can cause unexpectedly high space requirements, even with a short recovery window. Oracle recommends the use of a recovery window-based retention policy as part of a well-designed backup strategy.

Even after files in the flash recovery area are obsolete, they are generally not deleted from the flash recovery area until space is needed to store new files. As long as space permits, files recently moved to tape will remain on disk as well, so that they will not have to be retrieved from tape in the event of a recovery.

The automatic deletion of obsolete files and files moved to tape from the flash recovery area makes the flash recovery area a very convenient redo log archiving destination. Other archiving destinations require manual cleanup of archived redo logs no longer needed on disk for recovery.

Sizing the Flash Recovery Area

An approach for sizing the flash recovery area is described at length in *Oracle Database Backup and Recovery Basics*, but as a general rule, the larger the flash recovery area, the more useful it becomes. Ideally, the flash recovery area should be large enough to hold a copy of all of your datafiles and control files, the online redo logs, and the archived redo log files needed to recover your database using the datafile backups kept under your retention policy.

If your backup strategy includes incremental backups, which are described in "Incremental Backups of Datafiles" on page 9-10, then add enough space to the flash recovery area to accommodate these files as well. If you can move some backups to tape, then you can reduce the size of the flash recovery area somewhat. Note that retrieving those files from tape will cause longer database restore and recovery times.

Credentials for Performing Oracle Enterprise Manager Backup and Recovery

You must have the proper credentials to perform some of the configuration tasks for backup and recovery, and to schedule backup jobs and perform recovery. The following credentials may be required:

- The Oracle user you use when you log in to Enterprise Manager
- The host operating system user whose credentials you provide when performing backup and recovery tasks

To perform or schedule RMAN tasks, you must either log in to Enterprise Manager as a user with `SYSDBA` privileges, or provide host operating system credentials for a user who is a member of the `DBA` group. The host operating system user must also have execute permission for the RMAN command-line client.

For tasks requiring host operating system credentials, a Host Credentials form appears at the bottom of the page used to perform the task (see Figure 9–1, "Schedule Backup Page"). Enterprise Manager uses the credentials when it invokes RMAN to perform jobs you requested or scheduled.

Preferred Credentials for Backup and Recovery

The Host Credentials form always includes a checkbox labelled **Save as Preferred Credential**. If you check this box before performing your action, then the provided credentials are stored persistently for the currently logged-in Oracle user. The preferred credentials are reused by default whenever you log in as that user and perform operations requiring host credentials.

> **Note:** In situations in which the database is shut down, as is required for some database recovery operations, you may still be prompted for host credentials even if you save preferred credentials.

Configuring the Flash Recovery Area

You can configure a flash recovery area when first creating the database. If you did not perform this task at database creation time, however, then you can create a flash recovery area for your database now.

To configure a flash recovery area:

1. On the host operating system, create a directory to hold the flash recovery area. Make sure that the permissions for this directory allow Oracle to create files here.
2. From the Database Home page, click **Maintenance**.

   The Maintenance property page appears.
3. In the Backup/Recovery section, select **Recovery Settings**.

   The Recovery Settings page appears.

4. In the Flash Recovery section, enter the path to the flash recovery area location (the path to the directory on disk you created in step 1), and your desired flash recovery area size. Make sure the **Apply changes to SPFILE only** box is not checked, then click **Apply** to save your settings.

It is important to monitor space usage in the flash recovery area to ensure that it is large enough to contain backups and other recovery-related files. The High Availability section of the home page lists the percentage of flash recovery area space available. Click **Usable Flash Recovery Area** to navigate to the Recovery Settings page, which contains a Flash Recovery Area Usage graph showing how much space is allocated to each type of file and how much space is free.

Configuring ARCHIVELOG Mode For the Database

If you did not configure `ARCHIVELOG` mode when you first created your database, then configure it now with the following procedure. This procedure specifies that archived logs should be stored only in the flash recovery area, which is the best practice for managing archived redo logs.

To place the database in `ARCHIVELOG` mode:

1. On the Maintenance page, click **Recovery Settings**.

   The Recovery Settings property page appears.

2. In the Media Recovery section, check **ARCHIVELOG Mode** if it is not already checked. Under the **ARCHIVELOG Mode** checkbox is a list of up to ten possible log archiving locations. The last one is set to `USE_DB_RECOVERY_FILE_DEST`, indicating that the flash recovery area should be an archiving destination. You can specify other locations on disk as well.

   For ease of database management, the best practice is to use the flash recovery area as your only redo log archiving destination. Leave the other locations blank to store your archived logs in the flash recovery area only.

3. Click **Apply** to save your changes.

   If your database was not previously running in `ARCHIVELOG` mode, you are prompted to restart your database so that the switch to `ARCHIVELOG` mode can take effect.

4. Click **Yes** to indicate that you want to restart the database.

   The Restart Database:Specify Host and Target Database Credentials page appears.

5. Enter your credentials and click **OK**.

   The Restart Database:Confirmation page appears.

6. Click **Yes** to begin the restart process.

7. Make a consistent (that is, offline) backup of your whole database *immediately* after switching your database into `ARCHIVELOG` mode.

> **Caution:** Backups from before the switch to `ARCHIVELOG` mode cannot be used to restore and recover the database to a point in time after the switch. Thus, if you do not *immediately* take a backup after switching, you are running your database without a backup.
>
> See "Performing and Scheduling Backups with Enterprise Manager" on page 9-11 for more details on offline database backups.

Configuring Backup Settings

Assuming you have a flash recovery area configured and are running in ARCHIVELOG mode, you can configure a number of settings and policies that determine how backups are stored, which data is backed up, and how long backups are retained before being purged from the flash recovery area. You can also configure settings to optimize backup performance for your environment. This section provides information on concepts underlying the available settings, and information on how to change them through Enterprise Manager.

Understanding Backup Device Settings for Disk

The settings on the Device property page of the Backup Settings page affect how backups will be written to disk and to tape. For disk-based backups, you can configure the default format for storing backups, the location on disk where backups are stored, and whether backup tasks are run in parallel for improved performance.

Backup File Types Database backups created by RMAN can be stored in one of two forms: image copies or backup sets.

Image copies are exact byte-for-byte copies of the files they back up. You can create an image copy by copying a file at the host operating system level. Unlike copying files at the operating system level, however, creating image copy backups through RMAN or Enterprise Manager records those copies in the RMAN repository. This technique allows RMAN to use these copies during database restore and recovery. RMAN can only use files in restore operations if they are recorded in the RMAN repository. RMAN can only create image copies on disk.

Backup sets are logical entities containing backups produced by RMAN backup commands. Individual RMAN BACKUP commands can produce one or more backup sets. Each backup set consists of several physical files called backup pieces. A backup piece stores the backup of one or more database files in a compact format that can only be manipulated by RMAN. Backup pieces cannot be usefully manipulated individually; they can only be accessed as part of backup sets. Backup sets can be created on disk or media management devices such as tapes. Backup sets are the only form in which backups can be written to a media manager.

Parallelism and RMAN Backups RMAN depends on server sessions, processes that run on the database server, to perform backup and restore tasks. Each server session in turn corresponds to an RMAN channel, representing one stream of data to or from a backup device. RMAN supports parallelism, which is the use of multiple channels and server sessions to carry out the work of one backup or recovery task.

Proper exploitation of parallelism can greatly increase performance on backup and recovery tasks. For more conceptual material on channels and parallelism in RMAN backups, see *Oracle Database Backup and Recovery Advanced User's Guide*.

Configuring Backup Device Settings for Disk From the Database Home page, click **Maintenance**. In the Backup/Recovery section, choose **Backup Settings**.

The Backup Settings page contains three property pages: Device, Backup Set, and Policy. The settings you pick here are defaults which can apply to all backup jobs. When performing individual backup tasks, you can override these defaults.

The Device property page is shown first by default. Review the following fields under the Disk Settings section:

- **Parallelism**

For now, set this value to `1`. Later, when you have had time to review the information in *Oracle Database Backup and Recovery Advanced User's Guide* on parallelism and performance in RMAN, you may want to change this value.

- **Disk Backup Location**

  Leave this blank to direct backups to the flash recovery area.

- **Disk Backup Type**

  Make sure that **Backup Set** is selected. One advantage to backing up Oracle datafiles to backup sets is that RMAN uses unused block compression to save space in backing up datafiles. Only those blocks in your datafiles that have been used to store data are included in the backup set.

You can also provide the host credentials for the backup. Enter credentials as described in "Credentials for Performing Oracle Enterprise Manager Backup and Recovery" on page 9-6.

After these settings are filled in, you can click **Test Disk Backup** to make sure the credentials and backup location are correct.

The settings on the Backup Set property page should not be altered at this time.

Configuring Backup Policy Settings

From the Backup Settings page, click **Policy**. On the Policy property page, you can set the backup policies governing control file and SPFILE backups, tablespaces to exclude from whole database backup, and the backup retention policy.

Configuring Backup Policies Check the following options in the Backup Policy page:

- **Automatically backup the control file and server parameter file (SPFILE) with every backup and database structural change**

  The SPFILE and control file are critical to the operation of your database and RMAN and are also relatively small compared to typical datafiles. Backing them up frequently imposes relatively little storage overhead. Leave the **Autobackup Disk Location** field blank so that the autobackups are sent to the flash recovery area.

- **Optimize the whole database backup by skipping unchanged files such as read-only and offline datafiles that have been backed up**

  This option saves space in the flash recovery area.

- **Enable block change tracking for faster incremental backups**

  This option takes advantage of the block change tracking feature of Oracle, which substantially improves performance of incremental backups at a small cost of overhead during normal operations.

Configuring Exclusions from Backup You can specify a list of names for tablespaces to exclude from a backup. For example, you do not need to include read-only tablespaces in every backup. For now, make sure the list of excluded tablespaces is empty so that all tablespaces are backed up.

Configuring Backup Retention Policy You can choose among the following forms of retention policy:

- **Retain All Backups**

This option keep backups in the flash recovery area until you explicitly delete them. This option amounts to no retention policy at all.

- **Retain backups that are necessary for a recovery to any time within the specified number of days (point-in-time recovery)**

  This option enables a recovery window-based retention policy.

- **Retain at least the specified number of full backups for each datafile**

  This option enables a redundancy-based retention policy.

For now, choose the recovery window-based retention policy, with a recovery window of 31 days.

Verify that the **Host Credentials** section at the bottom of the page contains proper credentials. Click **OK** to save the new settings.

Backing Up Your Database

This section describes how to back up the database with Enterprise Manager. It introduces several types of Oracle database backup, then explains how to perform the different backup types, how to take advantage of Enterprise Manager's Oracle-suggested backup strategy to implement a useful basic backup regimen permitting fast recovery, and how to schedule your own backups.

> **Note:** The Oracle-suggested strategy for disk-only backups, as described in this section, provides efficient daily backup of your entire database to disk. This strategy enables you to quickly return your database to its state at any point during the preceding 24 hours. If you need more flexible backup options, then refer to *Oracle Database Backup and Recovery Basics*.

Database Backup Concepts

To understand the Oracle-suggested backup strategy and other backup types provided through Enterprise Manager, you need some conceptual background on database backups as supported by Oracle.

Full Backups of Datafiles

A full backup of a datafile is a backup that includes all used blocks of the datafile. This backup can be either an image copy backup, which is an exact copy of the datafile as if copied with a host operating system file copy command, or a backup set created by RMAN. Regardless of the form in which the backup is stored, the entire datafile is backed up, even if only a few blocks have changed.

Incremental Backups of Datafiles

Incremental backups capture only those blocks that change between backups in each datafile. In a typical incremental backup strategy, a level 0 incremental backup is used as a starting point. A level 0 backup captures all blocks in the datafile.

Subsequent level 1 incremental backups, typically made at regular intervals such as once each day, capture images of each block in a datafile that changed. Level 1 backups can be cumulative, in which case all blocks changed since the most recent level 0 backup are included, or differential, in which case only those blocks changed since the most recent level 0 or level 1 incremental backup are included.

Recovering changed blocks from incremental backups is used to improve media recovery performance. Because an incremental level 1 backup captures the final contents of all datafile blocks changed during the period covered by the incremental, the recovery process can skip reapplying individual updates from the redo logs of that period and simply update each block with its final contents. The redo logs are only used for changes from the period not covered by level 1 incremental backups.

Incrementally Updated Backups: Rolling Forward Image Copies of Datafiles

The **incrementally updated backups** feature of Oracle lets you use one or more level 1 incremental backups with an older image copy backup of your datafiles. You can roll the old copy forward to the point in time of the last level 1 incremental backup. All blocks changed since the image copy was created are overwritten with their new contents as of the time of the last level 1 incremental backup. The effect is to roll the file forward in time, so that its contents are equivalent, for the purposes of database recovery, to an image copy full datafile backup made at the time of the last incremental level 1 backup.

Incorporating incrementally updated backups into your backup strategy shortens expected recovery times. The reason is that media recovery to the present time or a point in time in the recent past can begin at the time of the last level 1 backup applied rather than the time of the last full database backup. The Oracle-Suggested Strategy for backups is based upon incrementally updated backups.

Using Tags to Identify Backups

All RMAN backups, including incremental backups, can be labelled with a tag. A tag is a text string identifying that backup, either uniquely or as part of a group of backups. For instance, if you performed a weekly full database backup on Saturday nights, you could use the tag `FULL_SATURDAY` to identify all such backups. You can use these tags for referring to specific backups in RMAN commands; for example, you could issue a command to move the latest `FULL_SATURDAY` backup to tape.

Because you can use tags to refer to different groups of backups, you can create different routines in your overall backup strategy that do not interfere with each other. When you schedule a backup job and give the job a name, the job name is used to tag the backup.

Performing and Scheduling Backups with Enterprise Manager

Enterprise Manager lets you perform all of the different backup types supported by RMAN, and schedule the different backup jobs required by your backup strategy.

Performing a Whole Database Backup with Oracle Enterprise Manager

Whole backups of a database are based on backing up the entire contents of the database at the time of backup. Full backups of all datafiles are created. The results may be stored as image copies or as backup sets, but in either case the complete contents of all datafiles of the database are represented in the backup, as well as the control file, archived redo log, and server parameter file. With this set of files, you can perform complete recovery of the database.

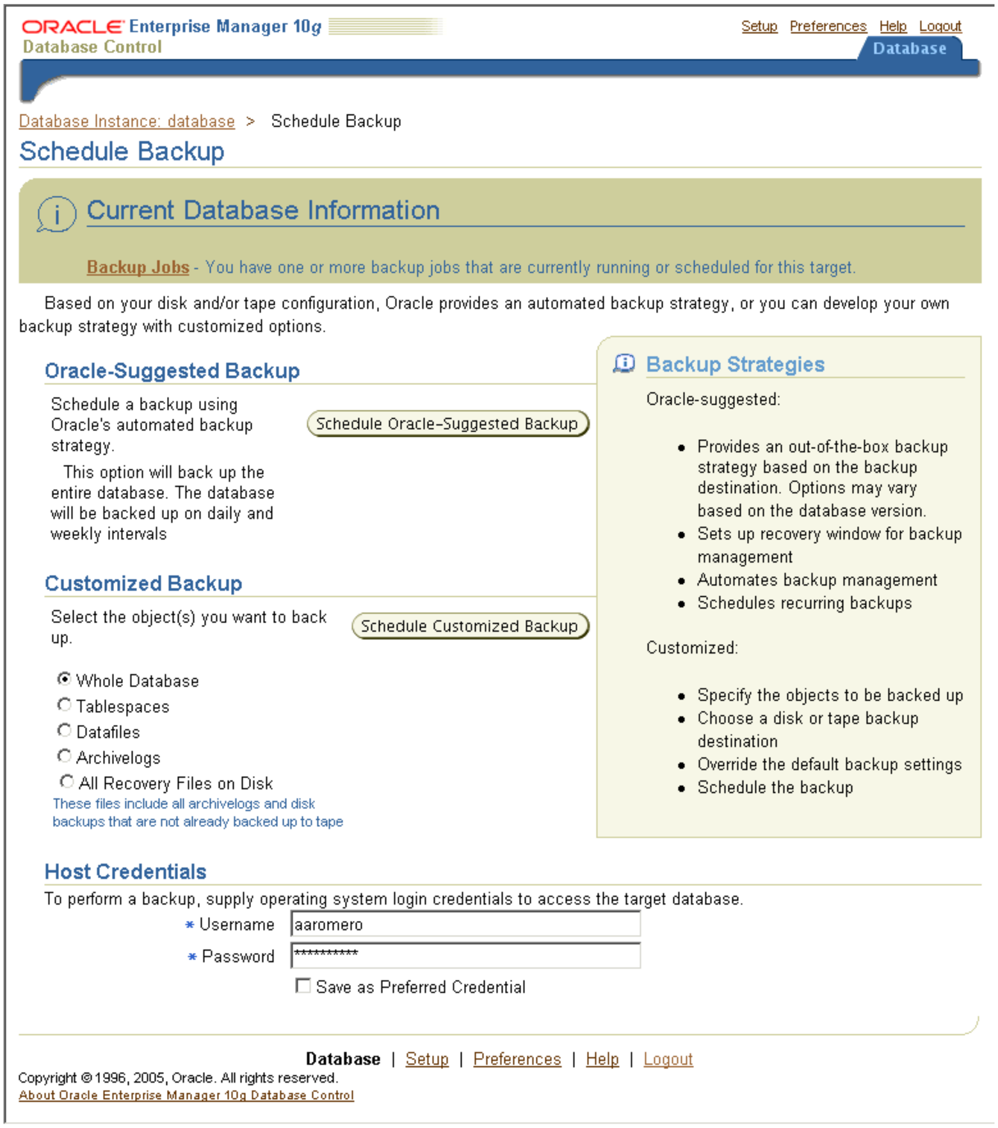
While whole database backups can be an important element in your overall backup strategy, they are also a required step in some situations, such as when you switch `ARCHIVELOG` mode on or off (see "Configuring ARCHIVELOG Mode For the Database" on page 9-7).

To perform a whole database backup, follow these steps:

1. In the Backup/Recovery section of the Maintenance page, click **Schedule Backup**.

   The Schedule Backup page appears, as shown in Figure 9–1.

Figure 9–1 Schedule Backup Page


ORACLE Enterprise Manager 10g
Database Control
Setup Preferences Help Logout
Database
Database Instance: database > Schedule Backup
Schedule Oracle-Suggested Backup
Schedule Customized Backup
Database | Setup | Preferences | Help | Logout
About Oracle Enterprise Manager 10g Database Control

Schedule Backup

Current Database Information

Backup Jobs - You have one or more backup jobs that are currently running or scheduled for this target.

Based on your disk and/or tape configuration, Oracle provides an automated backup strategy, or you can develop your own backup strategy with customized options.

Oracle-Suggested Backup

Schedule a backup using Oracle's automated backup strategy.

This option will back up the entire database. The database will be backed up on daily and weekly intervals

Customized Backup

Select the object(s) you want to back up.

● Whole Database
○ Tablespaces
○ Datafiles
○ Archivelogs
○ All Recovery Files on Disk

These files include all archivelogs and disk backups that are not already backed up to tape

Backup Strategies

Oracle-suggested:

- Provides an out-of-the-box backup strategy based on the backup destination. Options may vary based on the database version.
- Sets up recovery window for backup management
- Automates backup management
- Schedules recurring backups

Customized:

- Specify the objects to be backed up
- Choose a disk or tape backup destination
- Override the default backup settings
- Schedule the backup

Host Credentials

To perform a backup, supply operating system login credentials to access the target database.

* Username aaromero

* Password ***********

□ Save as Preferred Credential

Copyright © 1996, 2005, Oracle. All rights reserved.

The following sections of this page are important:

- The Oracle-Suggested Backup section, where you can click **Schedule Oracle-Suggested Backup** to choose among backup strategies recommended by the database
- The Customized Backup section, where you can schedule a one-time or repeated backup of database objects you select.

2. In the Customized Backup section, select **Whole Database** to make a full backup of your database immediately or schedule one as part of a user-designed backup strategy. Make sure the **Username** and **Password** fields are correct in the Host Credentials section and then click **Schedule Customized Backup**.

   The Schedule Customized Backup: Options page appears. In this page you specify the options for this whole database backup.

3. In the Backup Type section, select **Full Backup**. In the Backup Mode section, select either **Online Backup** or **Offline Backup**. Typically, you will want to perform online backups to maximize database availability.

   > **Note:** As explained in "Configuring Your Database for Basic Backup and Recovery" on page 9-4, you can only use online backups if your database is set up to run in `ARCHIVELOG` mode. In `NOARCHIVELOG` mode you can only perform offline backups.

   In the Advanced section, make the following selections:

   - Check **Also back up all archived logs on disk** if you are performing an online backup. There is no need to back up archived logs when performing an offline backup because the database is in a consistent state at the time of backup and does not require media recovery if you restore from this backup. Nevertheless, you can include archived logs in the backup if you wish.
   - Do not check **Delete all archived logs from disk after they are successfully backed up** if a flash recovery area is your only archiving destination. In this case, redo logs that have been backed up are deleted automatically as space is needed for storage of other files. If you are using some other destination, it may be useful to check this option as part of managing backup storage.
   - Do not check **Use proxy copy supported by media management software to perform a backup** for now.
   - Do not check **Delete obsolete backups** if you use a flash recovery area for backup storage. In this case, obsolete backups are deleted automatically as space is needed for storage of other files. If you are using some other destination, it may be useful to check this option as part of managing backup storage.
   - Do not enter a value for **Maximum Files per Backup Set**.

   After making your selections, click **Next**.

   The Schedule Customized Backup: Settings page appears.

4. Select a backup destination. Oracle recommends backing up to disk when possible to minimize recovery time by minimizing restores from tape. Backups created on disk can be moved to tape later. Click **Next**.

   The Schedule Customized Backup: Schedule page appears.

5. Specify identifying information for the backup job (including a tag and a description for your reference), and specify when the tasks for this backup job are performed:

   - In the Job section of the page, you can enter values for **Job Name** and **Job Description**. Default values for the name and description are generated for you. If you want to provide a tag for this backup, however, then enter the desired tag in the **Job Name** field. The job name is used as a prefix for the backup tag for backups produced by this job.

     You can set **Job Description** to any descriptive text that is useful for your own reference.
   - In the Schedule section, specify when to start the backup and how often to repeat it. Leave the default start time of **Immediately** selected to run a backup immediately, or set **Later** and enter a time in the future. For recurring backup

jobs, select options in the Repeat and Repeat Until sections. For one-time backups, select **One Time Only** and **Indefinite**.

> **Note:** For recurring jobs, it is useful to set a **Job Name** so that the resulting backups are easy to identify as part of an ongoing series. Use a descriptive tag, such as `WEEKLY_FULL_BACKUP`.

When finished, click **Next**.

The Schedule Customized Backup: Review page appears. This page presents a complete description of the backup job you specified in the previous pages.

6. You can perform one of the following actions:
   - Click the **Back** button to change these options.
   - Click **Edit RMAN Script** to view and, if desired, edit the RMAN commands that will be executed to perform your specified backup jobs.
   - Click **Submit Job** to add the specified backup job to the schedule (or to run it immediately, if you specified that the job should run immediately).
   - Click **Cancel** to stop the scheduled backup.

   In this example, click **Submit Job**.

   The Status page appears. This page should contain a message indicating that the job was successfully submitted.

7. Click **View Job** to monitor the progress of the backup job.

   The Execution page appears. This page contains a Summary section describing the job. The Logs section includes a table listing the progress of the various steps of the backup job. You can reload this page in your browser to monitor the ongoing progress of the job. In the Name column of the table in the Logs section, you can see what phase the RMAN job is in. Clicking on the name of the phase of the backup displays a page containing the RMAN output for that part of the job. From this page, click the **Back** button in your browser to return to the Execution page.

Performing Offline Database Backups When performing an offline backup, the database instance shuts down, then restarts and enters a `MOUNTED` state for the duration of the offline backup. The offline backup runs in the background, generating no user-visible output in the browser. The fact that the database is not open affects the pages you see from Enterprise Manager while the offline backup runs.

After you submit the backup job, a status page should appear indicating that the job has been successfully submitted. The output also includes a notification that the database will be shut down and mounted as part of the offline backup, and that you must wait for the backup to complete.

When the database is shut down and restarted, the Enterprise Manager application must also shut down for a brief time. During the period when Enterprise Manager is shut down, it cannot respond to attempts to refresh the page.

After Enterprise Manager restarts but the database is not open, Enterprise Manager reports that it cannot connect to the instance. The Database Instance section of the page reports the current state of the database listener and the instance (unmounted or mounted) as the database performs the offline backup. It also offers the options of **Startup** or **Perform Recovery**.

During the offline backup, do not click **Startup** or **Perform Recovery** because these may interfere with the offline backup. Instead, continue to refresh the page until the offline backup is complete and the database is restarted. At that time, Enterprise Manager prompts for login credentials. After you log in, you can return to the database home page.

Using the Oracle-Suggested Backup Strategy

Enterprise Manager makes it easy to set up an Oracle-suggested backup strategy for backups to disk that protects your data and provides efficient recoverability to any point in the a recovery window of your choosing. (In the simplest case, examined in this section, this window is 24 hours.) The Oracle-suggested strategy leverages Oracle's incremental backup and incrementally-updated backup features to provide faster backups than whole database backups, and faster recoverability than is possible through applying database changes from the archived log to your datafiles.

The Oracle-suggested backup strategy is based on creating an image copy of your database. This copy is rolled forward by means of incrementally updated backups. Oracle Enterprise Manager schedules RMAN backups jobs for you to run during the overnight hours.

For each datafile, the strategy calls for backups as follows:

- At the beginning of day 1 of the strategy (the time the first scheduled job actually runs), an incremental level 0 datafile copy backup. It contains the datafile's contents at the beginning of day 1.

  In a restore-and-recovery scenario, the redo logs from day 1 can be used to recover to any point during day 1.

- At the beginning of day 2, an incremental level 1 backup is created, containing the blocks changed during day 1.

  In a restore-and-recovery scenario, this incremental level 1 can be applied to quickly roll forward the level 0 backup to the beginning of day 2, and redo logs can be used to recover to any point during day 2.

- At the beginning of each day *n* for days 3 and onwards, the level 1 backup from the beginning of day *n*-1 is applied to the level 0 backup. This brings the datafile copy to its state at the beginning of day *n*-1. Then, a new level 1 is created, containing the blocks changed during day *n*-1.

  In a restore-and-recovery scenario, this incremental level 1 can be applied to the datafile rolled forward on day *n*-1 to the beginning of day *n*, and redo logs can be used to recover the database to any point during day *n*.

The datafile copies used in the Oracle-suggested backup strategy are tagged with the tag `ORA$OEM_LEVEL_0`. The level 1 incremental backups for use in this strategy are created for use with datafile copies that are so labelled. You can safely implement other backup strategies without concern for interference from the backups for the Oracle suggested strategy.

There are also Oracle-suggested strategies that use tape backups along with disk backups, but those are beyond the scope of this chapter.

Backing Up Your Database With the Oracle-Suggested Disk Backup Strategy Follow these steps to use the Oracle-suggested strategy to back up to disk:

1. In the Backup/Recovery section of the Maintenance page, click **Schedule Backup**.

   The Schedule Backup page appears, as shown in Figure 9–1.

2. In the Oracle-Suggested Backup section, click the **Schedule Oracle-Suggested Backup** button.

   The Schedule Oracle-Suggested Backup:Destination page appears. On this page, you select the destination media for the backup, which can be disk, tape or both.

3. Select **Disk** as the destination and click **Next**.

   The Schedule Oracle-Suggested Backup: Setup page. This page describes the backups that are performed each day as part of the disk-based strategy.

4. There are no settings to change on this page. Click **Next**

   The Schedule Oracle-Suggested Backup: Schedule page appears.

5. You are prompted for **Start Date**, **Time Zone**, and **Daily Backup Time** for the daily backups. Based upon your expected usage patterns, choose times for the nightly backup during which database activity is low. Click **Next**.

   The Schedule Oracle-Suggested Backup: Review page appears.

6. The backup script RMAN will run is displayed (although you cannot edit the script directly). You are presented with a chance to confirm or alter your settings. In the backup script, you can see the tag `ORA$OEM_LEVEL_0` that the script assigns to the backup. Assuming you do not need to change the schedule, click **Submit Job** to add the job for the Oracle-suggested strategy to your schedule.

   Your database will now be backed up once daily, using incremental backups and incrementally applied backups, allowing quick recovery to any time in the preceding 24 hours.

Scheduling Other Backup Tasks

After taking some time to understand the full range of available backup options as described in *Oracle Database Backup and Recovery Basics*, you may decide to schedule backup tasks beyond those used to implement the Oracle-suggested backup strategy.

While the particulars of specifying the job to perform differ for each type of backup, all backups begin from the Schedule Backup page shown in Figure 9–1. Here you can select any of several object types to back up. You can also back up existing backups in one destination to some other destination, such as moving backups from disk to tape.

Click **Schedule Customized Backup** to continue to the pages where you specify details such as objects to back up, required options and settings. The choices presented on these pages are determined by the type of objects being backed up. On each page, after you have made your selections, click Next to proceed to the next page.

After you have specified the options, you reach the Schedule Customized Backup: Schedule page, where you specify the times at which the job is to be performed, as well as the **Job Name** and Job Description.

When you are done with the job name, description and schedule, click Next to move on to the Schedule Customized Backup: Review page. After reviewing the options, either click **Back** to make changes, click **Edit RMAN Script** to make changes to the script, or click **Submit Job** to add the job to the schedule.

Validating Backups and Testing Your Backup Strategy

As part of your backup strategy, you should periodically check whether your backups are intact and can be used to meet your recoverability objectives.

Through Enterprise Manager, there are two different ways to validate your backups:

- You can select specific backup sets or image copies in Enterprise Manager and request that they be validated. This form of validation can reveal that a particular backup on disk or tape has been lost or corrupted.
- You can specify database files to restore, and let RMAN select backups to use in restoring those files, as it would for a real database restore operation. This form of backup validation ensures that your available backups are sufficient to restore your database. For example, this operation can reveal that your backup strategy does not back up all of your tablespaces, or that the loss of a particular backup prevents you from restoring a certain tablespace.

Note: Validating backups stored on tape can be time-consuming because the entire backup is actually read from tape.

Validating that you can perform specific restore operations with your available backups is performed through the Perform Recovery pages, and is described in "Validating the Restore of Datafiles from RMAN Backup" on page 9-20. Validating specific backup sets and image copies on disk or tape is performed through the Manage Current Backups pages, and is described in "Validating the Contents of Backup Sets or Image Copies" on page 9-28. Both forms of validation can be set up as scheduled tasks in Enterprise Manager. You should incorporate both forms of validation into your backup strategy to ensure that your recoverability goals are always met by your available backups.

Performing Restore and Recovery Operations

Enterprise Manager's Guided Recovery capability provides a Recovery wizard that encapsulates the logic required for a wide range of restore and recovery scenarios, including the following:

- Complete restore and recovery of the database
- Point-in-time recovery of the database or selected tablespaces
- Flashback Database
- Other flashback features of Oracle for logical-level repair of unwanted changes to database objects
- Block media recovery of datafiles with corrupt blocks

Enterprise Manager can determine which parts of the database must be restored and recovered, including proactively detecting situations such as corrupted database files. Enterprise Managers walks you through the recovery process, prompting for any needed information and performing needed recovery actions.

The examples in this section only illustrate a few common restore and recovery tasks. Nevertheless, the same Perform Recovery page is used to access the other whole database or object-level recovery features of Enterprise Manager.

To access the restore and recovery tasks:

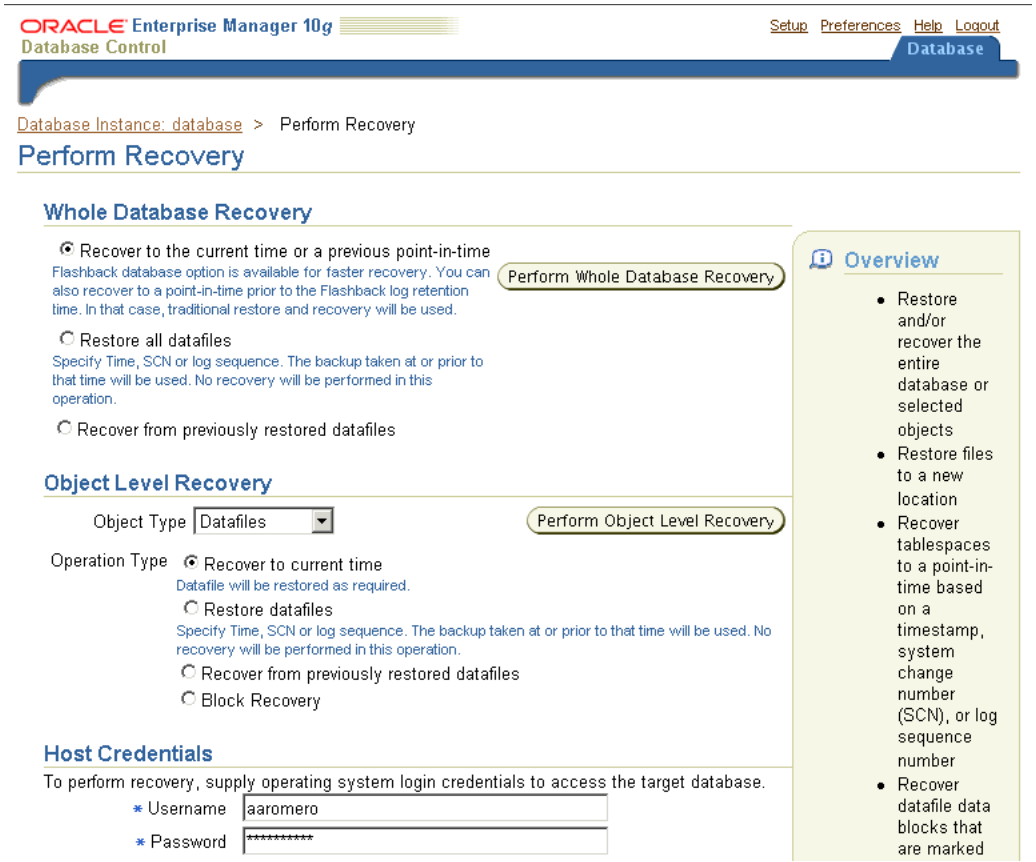
1. In the Database home page, click **Maintenance**.

   The Maintenance property page appears.

2. In the Backup/Recovery section of the Maintenance page, click **Perform Recovery**.

   The Perform Recovery page appears. A section of this page is shown in Figure 9–2.

Figure 9–2 Perform Recovery Page



Database Instance: database > Perform Recovery

Perform Recovery

Whole Database Recovery

● Recover to the current time or a previous point-in-time
Flashback database option is available for faster recovery. You can also recover to a point-in-time prior to the Flashback log retention time. In that case, traditional restore and recovery will be used.

○ Restore all datafiles
Specify Time, SCN or log sequence. The backup taken at or prior to that time will be used. No recovery will be performed in this operation.

○ Recover from previously restored datafiles

Object Level Recovery

Object Type Datafiles

Operation Type ● Recover to current time
Datafile will be restored as required.
○ Restore datafiles
Specify Time, SCN or log sequence. The backup taken at or prior to that time will be used. No recovery will be performed in this operation.
○ Recover from previously restored datafiles
○ Block Recovery

Host Credentials

To perform recovery, supply operating system login credentials to access the target database.

* Username aaromero
* Password ***********

Overview

- Restore and/or recover the entire database or selected objects
- Restore files to a new location
- Recover tablespaces to a point-in-time based on a timestamp, system change number (SCN), or log sequence number
- Recover datafile data blocks that are marked

On the Perform Recovery page, you can recover your whole database or only selected tablespaces, datafiles, archived logs, or tables.

Note: In some recovery scenarios, such as a complete restore and recovery of your database, the database state will be altered by steps you take during the wizard. Changes, some of them irreversible, are made to your database at certain steps. For example, the database may be shut down and brought to MOUNTED state, or datafiles may be overwritten with versions from backup.

Oracle Enterprise Manager will display warnings each time a significant database change will result from pressing **Continue** during the recovery process. Pay close attention to these warnings.

Recovering a Whole Database from Backup

This example demonstrates recovery of the whole database from backup. The example assumes that you are restoring and recovering your database after the loss of one or more datafiles, but still have a usable SPFILE and control file. Enterprise Manager can also be used to restore a lost SPFILE or control file. See "Recovering from a Lost SPFILE or Control File" on page 9-20 for details.

1. In the Backup/Recovery section of the Maintenance page, click **Perform Recovery**.

   The Perform Recovery page appears.

2. Select **Recover to the current time or a previous point-in-time** and click **Perform Whole Database Recovery**. Also, provide the requested host credentials at this time if necessary. Click **Continue**.

   The Confirmation page appears.

3. Click **Yes** to confirm the shutdown of the database.

   The Recovery Wizard page appears. At this point, Oracle shuts down your database.

   > **Note:** When the database is shut down and brought to the MOUNTED state, Enterprise Manager is also shut down briefly and restarted. During this process, there is a period during which Enterprise Manager cannot respond to your browser, or may respond with an error. Refresh the page until Enterprise Manager responds again.
   >
   > When Enterprise Manager has restarted and the database is being started and brought to the MOUNTED state, Enterprise Manager may also briefly report that the database is reported to be in NOMOUNT state. You are offered the choices **Refresh**, **Startup**, and **Perform Recovery**. Refresh the page periodically until the Database Instance page reports that the database instance is mounted before proceeding.

4. Click **Perform Recovery** to resume your recovery session. You may be prompted for host and database credentials. Connect with SYSDBA role, or provide host credentials for a user in the DBA group.

   When the Perform Recovery page is displayed again, it shows that the database is in a MOUNTED state (as is required for this operation).

5. At this point, as you did before, under Whole Database Recovery, select **Recover to the current time or a previous point-in-time** and click **Perform Whole Database Recovery**.

   The Perform Whole Database Recovery: Point-in-time page appears.

6. Specify whether to recover all transactions to your database as of the present time (which is called complete recovery), or only transactions up through some point in time in the past (which is called point-in-time recovery).

   > **Note:** Point-in-time recovery is a recovery technique that lets you return the database to its state before some unwanted major change. For more details about point-in-time recovery, see *Oracle Database Backup and Recovery Basics*.

   For this example, select **Recover to the current time** and then click **Next**.

   The Perform Whole Database Recovery: Rename page appears.

7. You can specify a new directory or new filename for the restored files. For this example, select **No** to restore the files to the default location, which is its location before the restore operation. Click **Next** to continue.

   The Perform Whole Database Recovery: Review page appears.

8. Review the options you chose. You can click **Edit RMAN script** to display the RMAN script that will be run to carry out your requested restore and recovery action. Click **Submit** to start the recovery.

Recovering from a Lost SPFILE or Control File

Recovery from loss of the control file or SPFILE must be started with the database shut down. Note that the instance cannot be running if the control file is lost.

Typically, diagnosis of the lost control file occurs when you attempt to start the database and the startup fails. The basic steps are as follows:

1. From the database home page, click **Startup** to attempt to start the database.
2. If the startup fails, click **View Details** to view the reason.
3. If the cause is a failure to locate the SPFILE or control file, return to the database home page and click **Perform Recovery**.
4. From this point, Enterprise Manager's Guided Recovery walks you through the process to restore the lost files.

After the control file is restored from backup, the database is mounted. When a control file restored from backup is used to start the database, you must perform complete recovery of the datafiles, even if none of the datafiles have been restored from backup. The database must be opened with the `RESETLOGS` option after the datafiles are recovered.

To perform complete recovery of the datafiles, use the following steps:

1. On the Perform Recovery page, select **Recover to the current time or a previous point in time** and click **Perform Whole Database Recovery**.
2. On the Perform Whole Database Recovery: Rename page, select **Restore files to the default location** and click **Next**.
3. On the Perform Whole Database Recovery: Review page, click **Submit** to start the media recovery process.

When recovery is complete, you are prompted to open the database with the `RESETLOGS` option.

Validating the Restore of Datafiles from RMAN Backup

Validating the restore of datafiles from a backup tests whether a sufficient set of backups exists that can be used to restore the specified files. After you specify which tablespaces to restore and, possibly, a point in time as of which to restore them, RMAN selects a set of backups that contain the needed data. RMAN then reads the selected backups in their entirety to confirm that they are not corrupt.

Note: This operation corresponds to the `RESTORE ... VALIDATE` command in Recovery Manager. See *Oracle Database Backup and Recovery Basics* for more information.

Validating the restore of files tests whether the file can be restored given the available backups, but it does not test whether all backups of the specified object are valid.

Validating particular backups and validating specific restore tasks are both useful in validating your backup strategy. For more information, see "Validating Backups and Testing Your Backup Strategy" on page 9-16.

1. In the Backup/Recovery section of the Maintenance property page, click **Perform Recovery**.

   The Perform Recovery page appears.

2. Specify the datafiles to validate individually or you can specify entire tablespaces. In the Object Level Recovery section, select **Datafiles** or **Tablespaces**. For the Operation Type, select **Restore Datafiles** or **Restore Tablespaces**. Make sure the host credentials are correct, and click **Perform Object Level Recovery**.

   The Perform Object Level Recovery page appears.

3. Click **Add** to add tablespaces or datafiles for the validate operation. After making your selections, click **Next**.

   The Perform Object Level Recovery: Restore page appears.

4. In the Backup Selection section, specify which backups to restore from. In the Backup Validation section, be sure to check **Validate the specified backup without restoring the datafiles**. Then click **Next**.

   The Perform Object Level Recovery: Schedule page appears.

5. Specify a job name and description. The job will run immediately, so you are not prompted for a scheduled time. Click **Next**.

   The Perform Object Level Recovery: Review page appears.

6. You can edit the RMAN script to be run or leave it as-is. Click **Submit Job** to run the validation.

   The Perform Recovery: Result page appears.

7. Click **View Job** to view the progress of the running job, or click **OK** to return to the database home page.

Returning a Table to a Past State: Flashback Table

Oracle Flashback Table lets you revert one or more tables back to their contents at a previous time without affecting other objects in your database. This recovery technique lets you recover from logical data corruptions, such as erroneously inserting rows into a table or deleting data from a table. Flashback Table lets you return tables you select to their state at a past point in time without undoing desired changes to the other objects in your database, as would be required by a point-in-time recovery of the entire database. Also, unlike point-in-time recovery, your database remains available during the operation.

For this example, you will perform Flashback Table on the `employees` table in the `hr` schema. Assume that an erroneous update shortly after October 23, 2005, 15:30:00 has changed the `lastname` column for all employees to an empty string and you need to return the original `lastname` values to the table.

Before you can perform Flashback Table, you must ensure that row movement is enabled on the table to be flashed back.

Enabling Row Movement on a Table

To enable row movement on a table, or if you do not know whether row movement is enabled on the table, follow these steps:

1. In the Database Objects section of the Administration page, click **Tables** to administer tables.

   The Tables page appears.

2. To find the target table for flashback table, you can enter one or both of the schema name in the **Schema** field and the table name in the **Object Name** field. Then click **Go** to search for the table. For example, search for tables in the `hr` schema. You may need to page through the search results to find your table.

3. After you find your table in the schema, select the table from the list of tables. For example, select `employees`. Click **Edit**.

   The Edit Table: *table_name* page appears.

4. Click **Options** to navigate to the Options property page. Make sure **Enable Row Movement** is set to **Yes**, and click **Apply** to update the options for the table.

When the page has refreshed, you can click **Tables** in the locator link at the top of the page to return to the search results, and enable row movement on more tables by repeating these steps for each table.

Performing Flashback Table

To perform the Flashback Table operation:

1. In the Backup/Recovery section of the Maintenance page, select **Perform Recovery**.

   The Perform Recovery page appears.

2. In the Object Level Recovery section, select **Tables** for the object type. The page reloads with options appropriate for object level recovery of tables. Choose the **Flashback Existing Tables** option and click **Perform Object Level Recovery**.

   The Perform Object Level Recovery: Point-in-time page appears.

3. Choose the target time for your Flashback Table operation.

   > **Note:** If you do not know the time at which the unwanted changes occurred, you can investigate the history of transactions affecting this table by choosing **Evaluate row changes and transactions to decide upon a point in time**. A feature called Oracle Flashback Versions Query lets you review all recent changes to the target table. Use of this feature is beyond the scope of this manual.

   For this example, assume that the time of the corruption is known to be October 23, 2005, 15:36:00. In the form offered, select **Flashback to a timestamp**, and enter your target time. Click **Next** to continue with the Flashback Table process.

   The Perform Object Level Recovery: Flashback Tables page appears.

4. Specify the target tables for Flashback Table by entering table names (one on each line) in the **Tables to Flashback** text box. You can enter multiple table names to flash several tables back to the same time. You can also click **Add Tables** and search for more tables to add. For this example, manually enter the `hr.employees` table in the **Tables to Flashback** text box. Click **Next** to continue with the Flashback Table process.

   If your table has other dependent tables, then the Dependency Options page appears. This page asks how dependencies should be handled.

5. You can choose **Cascade** (flashing back any dependent tables), **Restrict** (flashing back only the target table), or **Customize** (selecting which dependent tables to flashback and which to leave as they are). You can click **Show Dependencies** to

see which tables will be affected. How you proceed at this point will depend upon your application.

`hr.employees` has dependent tables `hr.jobs` and `hr.departments`. For this example, assume that it is safe to cascade any changes, flashing back those two tables as well as the `hr.employees` table.

> **Note:** Row movement must be enabled on all affected tables, not just the initial target tables.

Click **Next** to continue.

The Perform Object Level Recovery: Review page appears.

6. Review the target timestamp and tables to be flashed back. Click **Submit** to perform the Flashback Table operation.

   When the operation is completed, a Confirmation page displays the results. Click **OK** to return to the database home page.

Recovering Dropped Tables: Flashback Drop

Oracle Flashback Drop lets you reverse the effects of dropping a table, returning the dropped table to the database along with its dependent objects such as indexes and triggers. It works by storing dropped objects in a Recycle Bin, from which they may be retrieved until the Recycle Bin is purged, either explicitly or because space is needed for new database objects.

As with Flashback Table, Flashback Drop can be used while the rest of your database remains open, and without undoing desired changes in objects not affected by the Flashback Drop operation. It is more convenient than forms of recovery that require taking the database offline and restoring files from backup.

> **Note:** For a table to be recoverable using Flashback Drop, it must reside in a locally managed tablespace. Also, tables in the `SYSTEM` tablespaces cannot recovered using Flashback Drop regardless of the tablespace type.

To perform the Flashback Drop operation:

1. In the Backup/Recovery section of the Maintenance page, select **Perform Recovery**.

   The Perform Recovery page appears.

2. In the Object Level Recovery section, select **Tables** for the object type. The page reloads with options appropriate for tables in the Object Level Recovery section. For the **Operation Type**, select **Flashback dropped tables** and click **Perform Object Level Recovery**.

   The Perform Object Level Recovery: Dropped Objects Selection page appears.

3. The Search form lets you search among the dropped objects in the Recycle Bin for the objects you want to recover. Provide values for one or both of the **Schema Name** and **Table** fields, and click **Go** to search.

   When the page refreshes, the Results section lists the objects matching your search. If you only see the Recycle Bin listed, then click the arrow next to the Recycle Bin to expand its contents by one level, showing dropped tables matching your search but not their dependent objects. You can further expand individual tables, or click **Expand All** to see all objects in the Recycle Bin, including both dropped tables and dependent objects such as indexes and triggers. For each table listed, you can click **View Content** in the Operation column to display its contents.To select one or more tables for Flashback Drop, click the checkbox next to each table.

> **Note:** When a table is retrieved from the Recycle Bin, all of the dependent objects for the table that are in the recycle bin are retrieved with it. They cannot be retrieved separately.

When you have selected all of the objects to restore, click **Next**.

The Perform Object Level Recovery: Rename page appears.

4. If needed, specify new names for any dropped objects you are returning to your database. The primary reason for renaming objects when you retrieve them from the recycle bin is if you have created new tables with the same names as tables being retrieved. If you need to rename some objects, then enter new names as needed in the **New Name** field in the list of tables being flashed back. Click **Next** to continue.

   The Perform Object Level Recovery: Review page appears. This page displays an impact analysis, showing the full set of objects to be flashed back, including the dependent objects, as well as the names they will have when the Flashback Drop operation is complete.

5. If you are satisfied with the changes listed in the review, click **Submit** to perform the Flashback Drop.

   A confirmation page should indicate the success of the operation.

6. Click **OK** to return to the database home page.

Managing Your Backups

Managing RMAN backups, with or without Enterprise Manager, consists of two tasks: managing the backups of your database that are stored on disk or tape, and managing the record of those backups in the RMAN repository. Enterprise Manager simplifies both backup management tasks.

Backup Management: Concepts

A backup recorded in the RMAN repository can be in one of the following states:

- Available, meaning that the backup is still present on disk or tape, as recorded in the repository
- Expired, meaning that the backup no longer exists on disk or tape but is still listed in the repository
- Unavailable, meaning that the backup is temporarily not available for data recovery operations (because, for example, it is stored on a tape that is stored off-site or a disk that is not mounted at the moment)

Backups can also be obsolete. An obsolete backup is one which, based on the currently configured retention policy, is no longer needed to satisfy data recovery goals.

Backup maintenance tasks that you can perform in Enterprise Manager include the following:

- Viewing details about your backups
- Crosschecking your repository, which checks whether backups listed in the repository exist and are accessible and marks as expired any backups not accessible at the time of the crosscheck
- Deleting the record of expired backups from your RMAN repository
- Deleting obsolete backups from the repository and from disk
- Validating backups, to ensure that a given backup is still available and has not been corrupted

> **Note:** If a backup no longer exists, immediately delete the record of that backup from the RMAN repository. Without an accurate record of available backups, you may discover that you no longer have complete backups of your database when you try to perform a database recovery operation.

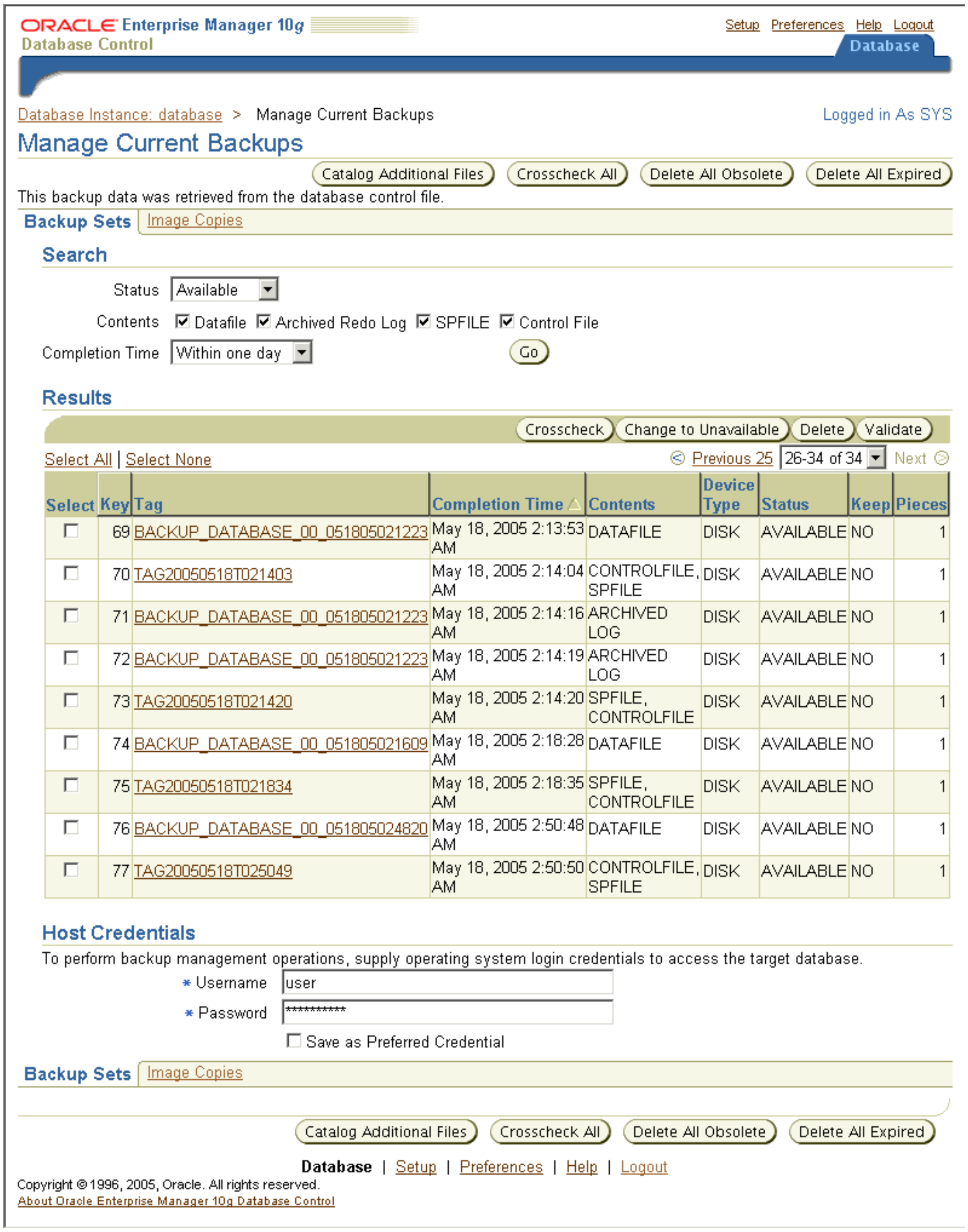
Note that if you use a flash recovery area for your backup storage, many maintenance activities are reduced or eliminated. The flash recovery area's automatic disk space management mechanisms delete backups and other files as needed, to satisfy demands for space in ongoing database operations, without compromising the retention policy.

Using the Manage Current Backups Page

To access backup management functions, navigate to the database home page and click **Maintenance** to open the Maintenance property page. In the Backup/Recovery section, click **Manage Current Backups**.

The Manage Current Backups page has two property pages you can choose: Backup Set and Image Copy. Each serves a similar purpose, listing backups (stored as backup sets or image copies), based on the record in the Recovery Manager repository, and allowing for management of the backups listed. Figure 9–3, "Manage Current Backups Page" shows a typical view of the Backup Set property page.

Figure 9–3 Manage Current Backups Page



Manage Current Backups

This backup data was retrieved from the database control file.

Search

Status: Available

Contents: ☑ Datafile ☑ Archived Redo Log ☑ SPFILE ☑ Control File

Completion Time: Within one day

Results

26-34 of 34

Select	Key	Tag	Completion Time	Contents	Device Type	Status	Keep	Pieces
□	69	BACKUP_DATABASE_00_051805021223	May 18, 2005 2:13:53 AM	DATAFILE	DISK	AVAILABLE	NO	1
□	70	TAG20050518T021403	May 18, 2005 2:14:04 AM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
□	71	BACKUP_DATABASE_00_051805021223	May 18, 2005 2:14:16 AM	ARCHIVED LOG	DISK	AVAILABLE	NO	1
□	72	BACKUP_DATABASE_00_051805021223	May 18, 2005 2:14:19 AM	ARCHIVED LOG	DISK	AVAILABLE	NO	1
□	73	TAG20050518T021420	May 18, 2005 2:14:20 AM	SPFILE, CONTROLFILE	DISK	AVAILABLE	NO	1
□	74	BACKUP_DATABASE_00_051805021609	May 18, 2005 2:18:28 AM	DATAFILE	DISK	AVAILABLE	NO	1
□	75	TAG20050518T021834	May 18, 2005 2:18:35 AM	SPFILE, CONTROLFILE	DISK	AVAILABLE	NO	1
□	76	BACKUP_DATABASE_00_051805024820	May 18, 2005 2:50:48 AM	DATAFILE	DISK	AVAILABLE	NO	1
□	77	TAG20050518T025049	May 18, 2005 2:50:50 AM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1

Host Credentials

To perform backup management operations, supply operating system login credentials to access the target database.

* Username: user

* Password: ***********

□ Save as Preferred Credential

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2005, Oracle. All rights reserved.

About Oracle Enterprise Manager 10g Database Control

The Image Copies property page is similar, with the same operations supported. The elements on the Manage Current Backups page can be categorized as follows:

- Buttons at the top of the page that allow you to perform maintenance tasks related to your entire collection of backups
- The Search section, which enables you to select a subset of your backup sets or image copies for maintenance actions
- The Results section, where you can select individual backups for maintenance actions, invoked using buttons at the top of the Results section

Searching for Backups on the Manage Current Backups Page

The Search section of each property page lets you restrict the listed backups based on the following:

- Backup status (available, unavailable, or expired)
- The types of files contained in the backup (archived log, datafile, control file, or SPFILE)
- How recently the backup was completed

By default, all available backups from the last month are listed. Specify criteria as necessary to filter the backup list, and click **Go** to search the backups.

The Results section of the page lists backups that match the criteria specified in the Search section. Different details are listed for backup sets (on the Backup Sets property page) and for image copies (on the Image Copies property page).

Managing Current Backups: Backup Sets

On the Backup Sets property page, backup sets are identified by the unique value in the Key column (an internally generated number used to identify the backup set in RMAN commands) as well as the value in the Tag column. For each backup, the page lists information such as the type of files backed up, the destination device type, completion time of the backup, and current status.

You can click the value in the Tag column for a given backup set to view details about the backup set, including:

- The input files backed up (listed in the Input Files section of the page), and details about each input file, including the original file location, size, and file checkpoint time and SCN when the backup was taken
- The backup pieces produced as output by the backup job, including the output filename, tag, destination device type, size, and status

Managing Current Backups: Image Copies

On the Image Copies property page, image copies are identified by the unique value in the Key column (an internally generated number used to identify the backup set in RMAN commands) as well as the filename in the Name column. For each backup, the file type, tag, completion time and status are also listed. This page also lists the status of each image copy (available, unavailable, or expired) and a Keep column, which shows whether special exceptions to the retention policy for backups apply to this file.

By clicking the value in the Name column, you can also view details about that file, such as datafile number, file size, tablespace name, creation SCN, and last checkpoint SCN and time when the file was backed up.

You can also crosscheck or delete individual backups, or mark individual backups as unavailable if you know that they are temporarily not accessible by RMAN. For example, if you know that some backups are stored on tapes that have been moved offsite for long-term storage, you can mark them as unavailable to prevent RMAN from considering them usable in restore operations. Use the **Select** checkbox next to the file, and click the appropriate action button at the top of the Results list.

The Image Copy property page presents similar functionality to the Backup Sets property page. The focus in this section is on commands in the Backup Sets property page, where they are substantially similar to those for image copies.

As with the backup and restore commands in Enterprise Manager, the commands to crosscheck, delete and change the status of backups are ultimately translated to

RMAN commands. The commands are submitted as RMAN jobs that can be run immediately or scheduled. Some tasks, such as periodic crosschecks of your backups, should be among the regularly scheduled components of your backup strategy.

Validating the Contents of Backup Sets or Image Copies

When you validate a backup, you ensure that the files for the backup are present, have not been corrupted, and can be used in recovery operations. The selected backup files are read in their entirety and checked for errors.

> **Note:** Validating specific backups only checks whether those backups are readable. It does not test whether the set of available backups meet your recoverability goals.
>
> For example, you can have image copies of datafiles for several tablespaces from your database, each of which can be validated. If there are some tablespaces for which no valid backups exist, however, you cannot restore and recover your database.
>
> Validating a restore operation ensures that the set of backups available can actually be used to perform a specified restore operation, such as restoring a specific tablespace or your entire database. For more information on validating restore operations, see "Validating Backups and Testing Your Backup Strategy" on page 9-16.

To validate whether specific backups are usable in a restore and recovery operation:

1. In the Backup/Recovery section of the Maintenance property page, click **Manage Current Backups**.

   The Manage Current Backups page appears.

2. Use the search features on this page to identify the backups whose contents you wish to validate, as described in "Using the Manage Current Backups Page" on page 9-25.

3. Select the checkboxes next to each backup of interest in the list of current backups and click **Validate**.

   The Validate: Specify Job Parameters page appears.

4. Specify a job name and description, as well as time settings for starting and repeating the operation. You can click **Show RMAN Script** to view the RMAN commands used to perform your validation or **Submit Job** to start the job running according to its schedule.

   A message confirms the submission of the job.

5. Click **View Job** to display details about the validation.

This process corresponds to the RMAN `VALIDATE` command. See *Oracle Database Backup and Recovery Basics* for details on the use of the RMAN `VALIDATE` command.

Crosschecking Backups

Crosschecking a backup causes RMAN to verify that the actual physical status of the backup matches the record of the backup in the RMAN repository. For example, if a backup on disk has been deleted with an operating system command and is therefore no longer available for use in restore operations, then crosschecking that file detects this condition. After the crosscheck operation, the RMAN repository correctly reflects the state of the backups on disk or tape.

Backups to disk are marked `AVAILABLE` if they are still present on disk in the location listed in the RMAN repository, and if they have no corruption in the file header. Backups on tape are listed as `AVAILABLE` if they are still present on tape (though the file headers are not checked for corruption). Backups that are missing or corrupt are marked `EXPIRED`.

To crosscheck individual files:

1. Use the search features of the Manage Current Backups page to find the backup sets or image copies whose contents you wish to crosscheck, as described in "Using the Manage Current Backups Page" on page 9-25.
2. Click the **Select** checkbox next to each backup in the Results list to be included in the crosscheck operation. Enterprise Manager does not support selecting both image copies and backup sets for crosscheck within a single operation.
3. Click **Crosscheck** at the top of the Results list. After a confirmation page, Enterprise Manager performs the crosscheck.

You can also crosscheck all backups recorded in the RMAN repository in one step by clicking **Crosscheck All** at the top of the page. In the Crosscheck All: Specify Job Parameters page, you can schedule the crosscheck to run now or at a later time, or even specify regularly scheduled crosschecks.

> **Note:** Crosschecking all backups in the RMAN repository may take a long time, especially for tape backups. A crosscheck of all files, unlike crosschecking individual files, is handled as a scheduled job.

This process corresponds to the RMAN `CROSSCHECK` command. See *Oracle Database Backup and Recovery Basics* for details.

Deleting Expired Backups

Deleting expired backups removes from the RMAN repository those backups which are marked `EXPIRED`, that is, those which RMAN found to be inaccessible during a crosscheck operation. No attempt is made to delete the files containing the backup from disk or tape; this action only updates the RMAN repository.

To delete expired backups:

1. In the Manage Current Backups page, click **Delete All Expired**. Note that this action deletes both expired backup sets and expired image copies from the RMAN repository, regardless of whether you are viewing the Backup Sets or Image Copies property page when you click **Delete All Expired**.

   The Delete All Expired: Specify Job Parameters page appears.

2. Select your scheduling options. Along with the usual scheduling options for an RMAN job, there is a checkbox **Perform the operation 'Crosscheck All' before 'Delete All Expired'**. Checking this box will cause the operation to take longer, but by performing the crosscheck operation immediately before deleting expired backups from the repository, RMAN will have the most up-to-date information possible about which backups are expired. Click **Submit Job**.

   A message should appear indicating that your job was successfully submitted.

3. Click **View Job** to display details about the deletion operation.

Marking Backups as Available or Unavailable

If you know that one or more specific backups are unavailable because of a temporary condition, such as a disk drive that is temporarily offline or a tape stored off-site, then you can mark those backups as unavailable. RMAN does not try to use unavailable backups in restore and recovery operations. RMAN keeps the record of unavailable backups in the RMAN repository, however, and when you delete expired backups, RMAN does not try to delete backups marked unavailable. When the backups become accessible again, you can mark them as available.

Included among the buttons at the top of the Results section of the Manage Current Backups page are buttons for marking backups as available or unavailable.

Note: If you have restricted the backups listed by searching only for available backups, only the **Change to Unavailable** button appears. If you have restricted the backups listed by searching only for unavailable backups, only the **Change to Available** button appears.

To mark backups as available, click the **Select** checkbox next to each backup in the Results list of backups, and select **Change to Available**.

To mark backups as unavailable, click the **Select** checkbox next to each backup in the Results list of backups, and select **Change to Unavailable**.

Note: Backups stored in the flash recovery area cannot be marked as unavailable.

Deleting Obsolete Backups

To delete obsolete backups, that is, backups no longer needed to meet your retention policy, click **Delete All Obsolete** at the top of the Manage Current Backups page.

When you click **Delete All Obsolete**, you arrive at the Delete All Obsolete: Specify Job Parameters page. You can run the deletion job immediately or schedule it as you would a backup job.

Note: All obsolete backups (both backup sets and image copies) will be deleted, regardless of whether you clicked **Delete All Obsolete** while viewing the Backup Set or Image Copy property page on the Manage Current Backups page.

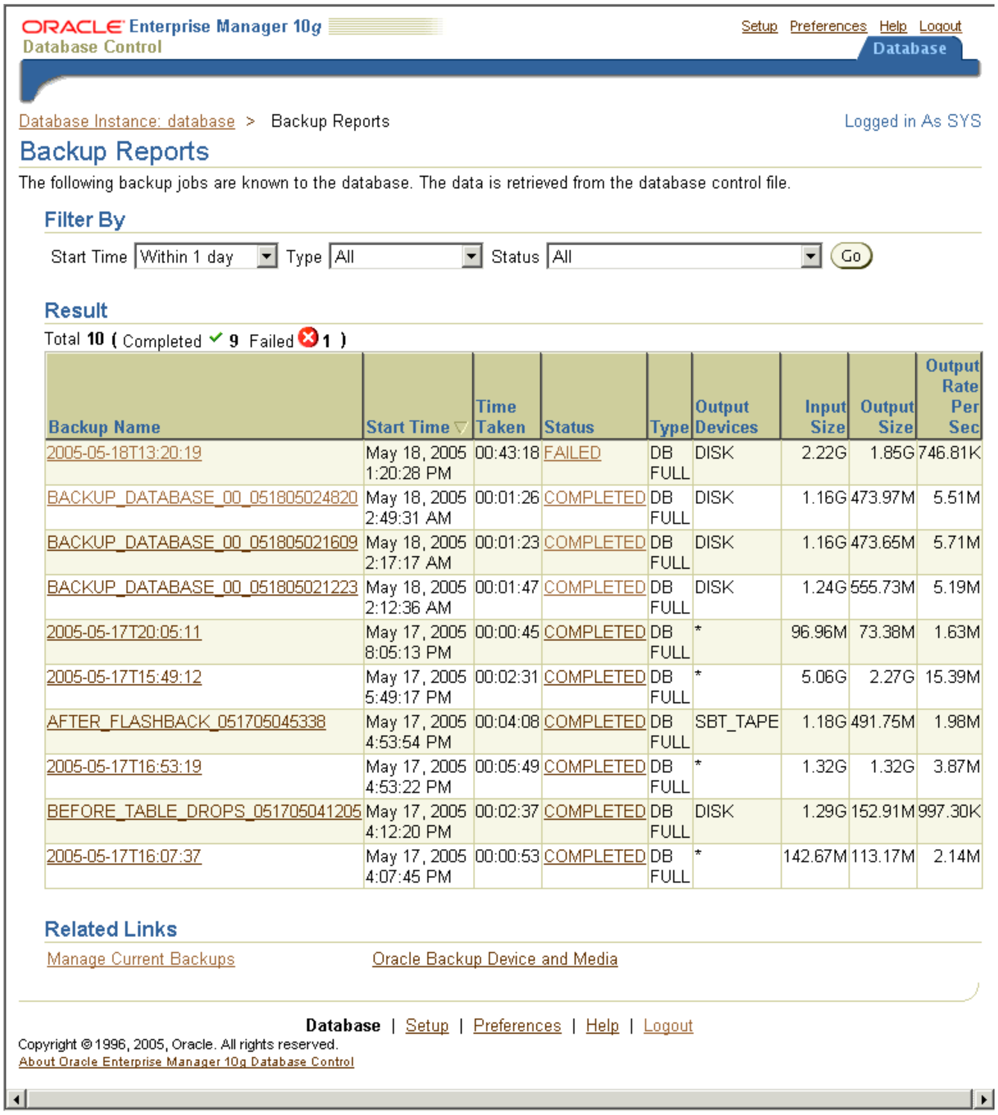
Note that if you use a flash recovery area as your sole disk-based backup destination, then you never need to delete obsolete backups from disk. The automatic space management of the flash recovery area will keep files as specified by the backup retention policy, and then only delete them when space is needed.

Displaying Backup Reports

Backup reports contain summary and detailed information about past backup jobs run by RMAN, including both backups run through Enterprise Manager and the RMAN command-line client.

To view backup reports, from the Maintenance property page, in the Backup and Recovery section, click **Backup Reports**.

Figure 9–4 Backup Reports Page



ORACLE Enterprise Manager 10*g*
Database Control

Setup Preferences Help Logout
Database

Database Instance: database > Backup Reports

Logged in As SYS

Backup Reports

The following backup jobs are known to the database. The data is retrieved from the database control file.

Filter By

Start Time: Within 1 day; Type: All; Status: All; Go

Result

Total **10** (Completed ✓ **9** Failed ✗ **1**)

Backup Name	Start Time ▽	Time Taken	Status	Type	Output Devices	Input Size	Output Size	Output Rate Per Sec
2005-05-18T13:20:19	May 18, 2005 1:20:28 PM	00:43:18	FAILED	DB FULL	DISK	2.22G	1.85G	746.81K
BACKUP_DATABASE_00_051805024820	May 18, 2005 2:49:31 AM	00:01:26	COMPLETED	DB FULL	DISK	1.16G	473.97M	5.51M
BACKUP_DATABASE_00_051805021609	May 18, 2005 2:17:17 AM	00:01:23	COMPLETED	DB FULL	DISK	1.16G	473.65M	5.71M
BACKUP_DATABASE_00_051805021223	May 18, 2005 2:12:36 AM	00:01:47	COMPLETED	DB FULL	DISK	1.24G	555.73M	5.19M
2005-05-17T20:05:11	May 17, 2005 8:05:13 PM	00:00:45	COMPLETED	DB FULL	*	96.96M	73.38M	1.63M
2005-05-17T15:49:12	May 17, 2005 5:49:17 PM	00:02:31	COMPLETED	DB FULL	*	5.06G	2.27G	15.39M
AFTER_FLASHBACK_051705045338	May 17, 2005 4:53:54 PM	00:04:08	COMPLETED	DB FULL	SBT_TAPE	1.18G	491.75M	1.98M
2005-05-17T16:53:19	May 17, 2005 4:53:22 PM	00:05:49	COMPLETED	DB FULL	*	1.32G	1.32G	3.87M
BEFORE_TABLE_DROPS_051705041205	May 17, 2005 4:12:20 PM	00:02:37	COMPLETED	DB FULL	DISK	1.29G	152.91M	997.30K
2005-05-17T16:07:37	May 17, 2005 4:07:45 PM	00:00:53	COMPLETED	DB FULL	*	142.67M	113.17M	2.14M

Related Links

Manage Current Backups   Oracle Backup Device and Media

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2005, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Database Control

The page contains a list of recent backup jobs. You can use the Filter By section of the page to restrict the backups listed by the time of the backup, the type of data backed up, and the status of the jobs to be listed (whether it succeeded or failed, and whether warnings were generated during the job). Specify any filter conditions and click **Go** to restrict the list to backups of interest.

The list contains basic information for each backup:

- The name, start time and run time of the backup
- The type of the backup (such as full or incremental database backup, archived log backup, backup of datafiles only, and so on)
- The status of the backup (whether it succeeded or failed, and whether there were warnings associated with it)
- The output destination (which can be disk or tape, or * to indicate output on both disk and tape
- The size of inputs (files backed up), output (files written to disk or tape) and the rate at which backups were written

To view detailed information about any backup, click the value in the **Backup Name** column. The View Backup Report page is displayed for the selected backup. This page contains summary information about this backup (how many files of each type were backed up, how much data total, and the number, size and type of output files created), as well as details about inputs and outputs for the backup job, such as:

- Datafiles backed up, by datafile number and tablespace name
- Archived logs by sequence and thread number, and range of times and SCNs covered by the logs
- Number of corrupt blocks found in input files
- Tags, backup piece names or filenames for outputs
- Size of output files and compression ratios

For recent jobs, you can also view a log of RMAN output for the job, by clicking the value in the **Status** column.

> **Note:** The control file view `V$RMAN_OUTPUT` contains the output of recent RMAN jobs. The contents of this view are not preserved if the instance is restarted. Therefore, the output from past RMAN jobs may not be available.

The **View Backup Report** page also contains a Filter By section that you can use to quickly run a search for another backup or backups from a specific date range. The resulting report contains aggregate information for backups matching the search criteria.

Backup and Recovery: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter, and includes annotated screen shots.

To view the Backup and Recovery OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/backup/backup.htm
```

10

Monitoring and Tuning the Database

Monitoring the health of a database and ensuring that it performs optimally is an important task for a database administrator. This chapter discusses the features and functionality included in Oracle Database that make it easy to proactively monitor database health, identify performance problems, and implement any corrective actions.

The following topics are covered:

- Proactive Database Monitoring
- Diagnosing Performance Problems
- Using Advisors to Optimize Database Performance
- Monitoring and Tuning: Oracle by Example Series

Proactive Database Monitoring

The Oracle Database makes it easy to proactively monitor the health and performance of your database. It monitors the vital signs (or metrics) related to database health, analyzes the workload running against the database, and automatically identifies any issues that need your attention as an administrator. The identified issues are either presented as alerts and performance findings in Enterprise Manager or, if you prefer, can be sent to you through e-mail.

This section discusses the following topics:

- Alerts
- Performance Self-Diagnostics: Automatic Database Diagnostics Monitor
- Monitoring General Database State and Workload
- Managing Alerts

Alerts

Alerts help you monitor your database. Most alerts notify you of when particular metric thresholds are crossed. For each alert, you can set critical and warning threshold values. These threshold values are meant to be boundary values that when crossed indicate that the system is in an undesirable state. For example, when a tablespace becomes 97 percent full this can be considered undesirable, and Oracle will generate a critical alert.

Other alerts correspond to database events such as Snapshot Too Old or Resumable Session suspended. These types of alerts indicate that the event has occurred.

In addition to notification, you can set alerts to perform some action such as running a script. For instance, scripts that shrink tablespace objects can be useful for a Tablespace Usage warning alert.

By default, Oracle enables the following alerts:

- Table Space Usage (warning at 85 percent full, critical at 97 percent full)
- Snapshot Too Old
- Recovery Area Low on Free Space
- Resumable Session Suspended

You can modify these alerts or enable others by setting their metrics.

For more information, see "Managing Alerts" on page 10-4.

Performance Self-Diagnostics: Automatic Database Diagnostics Monitor

Oracle Database includes a self-diagnostic engine called the Automatic Database Diagnostic Monitor (ADDM). ADDM makes it possible for the Oracle Database to diagnose its own performance and determine how any identified problems can be resolved.

To facilitate automatic performance diagnosis using ADDM, Oracle Database periodically collects snapshots of the database state and workload. The default collection interval is one hour. Snapshots provide a statistical summary of the state of the system at any point in time. These snapshots are stored in the Automatic Workload Repository (AWR), residing in the `SYSAUX` tablespace. The snapshots are stored in this repository for a set time (a week by default) before they are purged to make room for new snapshots.

ADDM analyses data captured in AWR to determine the major problems in the system and in many cases recommends solutions and quantifies expected benefits.

Generally, the performance problems ADDM can flag include the following:

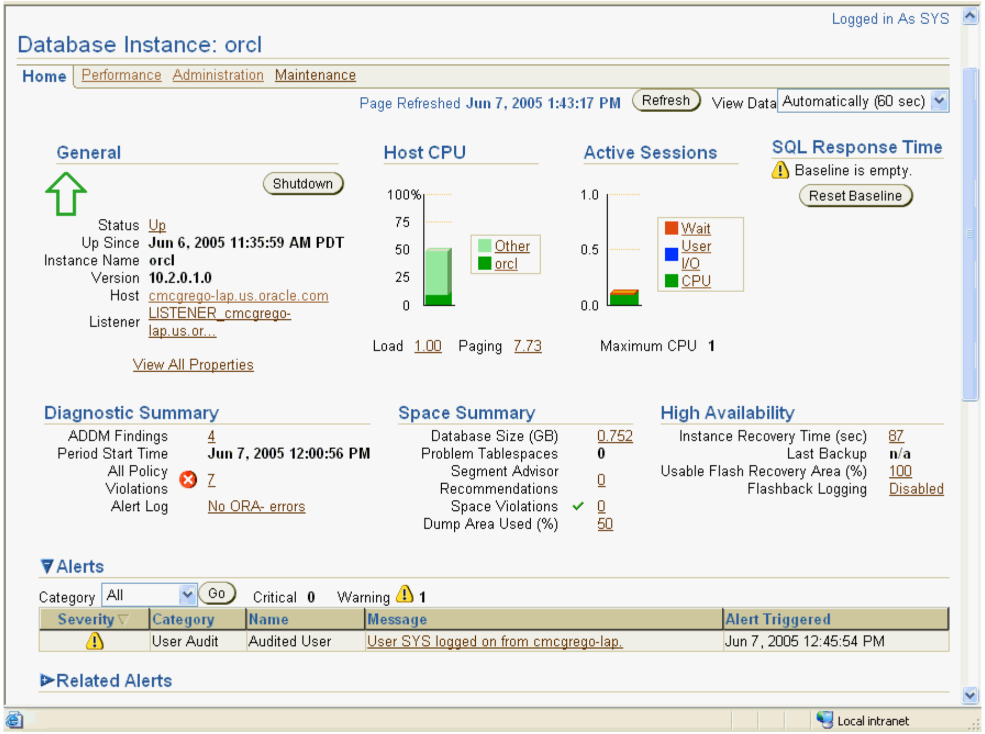
- Resource bottlenecks, such as when your database is using large amounts of CPU time or memory, for example as a result of high load SQL statements
- Poor connection management, such as when your application is making too many logons to the database
- Lock contention in a multiuser environment, when a lock to update data causes other sessions to wait, slowing down the database

> **See Also:** For more information about using ADDM, see "Diagnosing Performance Problems" on page 10-9 and the *Oracle Database Performance Tuning Guide*

Monitoring General Database State and Workload

The Enterprise Manager home page enables you to monitor the health of your database. It provides a central place for general database state information and is updated periodically. This page reports information that is helpful for monitoring database state and workload. See Figure 10–1, "Enterprise Manager Home Page".

Figure 10–1 Enterprise Manager Home Page



Database Instance: orcl

Page Refreshed **Jun 7, 2005 1:43:17 PM** View Data Automatically (60 sec)

General

Status Up
Up Since **Jun 6, 2005 11:35:59 AM PDT**
Instance Name **orcl**
Version **10.2.0.1.0**
Host cmcgrego-lap.us.oracle.com
Listener LISTENER_cmcgrego-lap.us.or...

Load 1.00 Paging 7.73

Maximum CPU **1**

SQL Response Time

⚠ Baseline is empty.

Diagnostic Summary

ADDM Findings 4
Period Start Time **Jun 7, 2005 12:00:56 PM**
All Policy Violations ☒ 7
Alert Log No ORA- errors

Space Summary

Database Size (GB) 0.752
Problem Tablespaces **0**
Segment Advisor Recommendations 0
Space Violations ✓ 0
Dump Area Used (%) 50

High Availability

Instance Recovery Time (sec) 87
Last Backup **n/a**
Usable Flash Recovery Area (%) 100
Flashback Logging Disabled

▼Alerts

Category All Critical **0** Warning ⚠ **1**

Severity ▽	Category	Name	Message	Alert Triggered
⚠	User Audit	Audited User	User SYS logged on from cmcgrego-lap.	Jun 7, 2005 12:45:54 PM

▶Related Alerts

The **General** section provides a quick view of the database, such as whether the database is Up or Down, the time the database was last started, instance name, and host name.

The **Host CPU** section shows the percentage of CPU time used in the overall system. This chart breaks down CPU percentage into time used by the database and time used by other processes. If your database is taking up most of the CPU time, you can explore the cause further by looking at the **Active Sessions** summary. This summary tells you what the database processes are doing, such as which ones are using CPU, or waiting on I/O. You can drill down for more information by clicking a link, such as **CPU**.

If other processes are taking up most of your CPU time, this indicates that some other application running on the database machine may be causing the performance problems. To investigate this further, click the **Host** link under the General section. This link takes you to the machine overview page where you can see some general information about the machine, such as what operating system it is running, how long the machine has been up, and any potential problems.

Click the **Performance** property page to see a summary of CPU utilization, average active sessions, instance disk I/O, and instance throughput. Additional Monitoring Links enable you to drill down to Top Activity and other data. The type of actions you can take to improve host performance depends on your system, but can include eliminating unnecessary processes, adding memory, or adding CPUs.

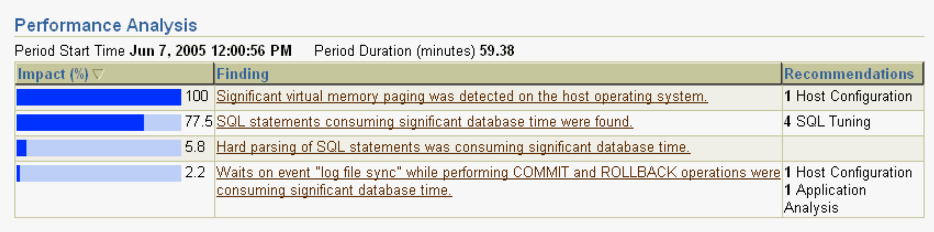
On the Home page, the **Diagnostic Summary** summarizes the latest ADDM performance findings. This section also summarizes any critical or warning alerts listed in the Alerts section.

The **Alerts** table provides information about any alerts that have been issued along with the severity rating of each. An alert is a notification that a metric threshold has been crossed. For example, an alert can be triggered when a tablespace is running out of space.

When an alert is triggered, the name of the metric causing it is displayed in the Name column. The severity icon (Warning or Critical) is displayed, along with time of alert, and alert value. You can click the message to learn more about the cause. For more information, see "Alerts" on page 10-1.

The **Performance Analysis** section provides a quick summary of the latest ADDM findings, highlighting the issues that are causing the most significant performance impact. This analysis can identify problems such as SQL statements that are consuming significant database time. See Figure 10–2, "Performance Analysis on Home Page".

Figure 10–2 Performance Analysis on Home Page



Performance Analysis

Period Start Time **Jun 7, 2005 12:00:56 PM** Period Duration (minutes) **59.38**

Impact (%) ▽	Finding	Recommendations
100	Significant virtual memory paging was detected on the host operating system.	**1** Host Configuration
77.5	SQL statements consuming significant database time were found.	**4** SQL Tuning
5.8	Hard parsing of SQL statements was consuming significant database time.	
2.2	Waits on event "log file sync" while performing COMMIT and ROLLBACK operations were consuming significant database time.	**1** Host Configuration **1** Application Analysis

For more information, see "Performance Self-Diagnostics: Automatic Database Diagnostics Monitor".

Managing Alerts

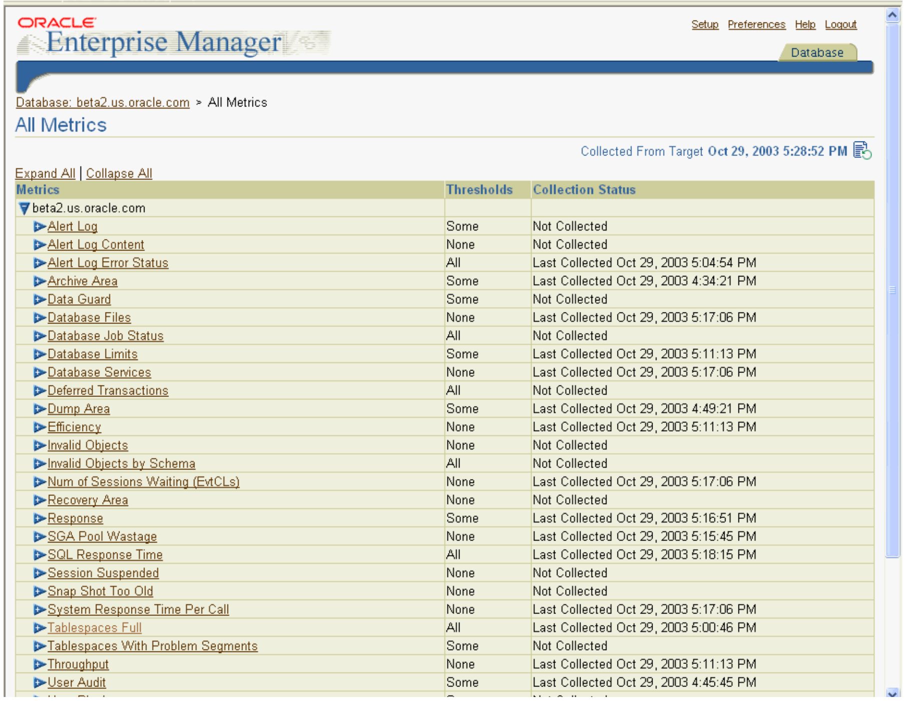
The following sections describe how to manage alerts.

Viewing Metrics and Thresholds

Metrics are a set of statistics for certain system attributes as defined by Oracle. They are computed and stored by the Automatic Workload Repository, and are displayed on the All Metrics page, which is viewable by clicking **All Metrics** under Related Links on the Database Home page.

Figure 10–3 shows a portion of the All Metrics page, displaying some of the metrics that AWR computes. When you click a specific metric link, a detail page appears, with more information about the metric. Online Help for this page gives you a description of the metric.

Figure 10–3 All Metrics Page



Database: beta2.us.oracle.com > All Metrics

All Metrics

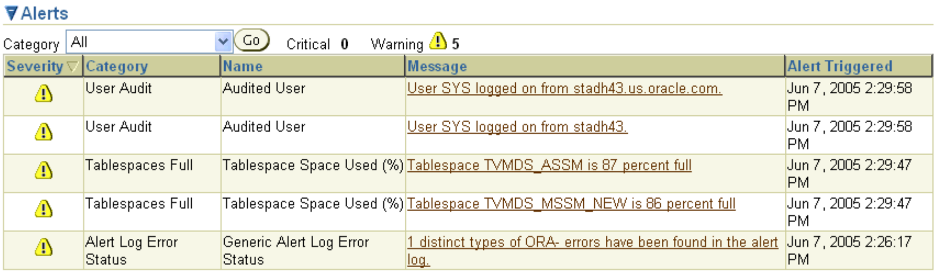
Collected From Target Oct 29, 2003 5:28:52 PM

Metrics	Thresholds	Collection Status
▼beta2.us.oracle.com		
▷Alert Log	Some	Not Collected
▷Alert Log Content	None	Not Collected
▷Alert Log Error Status	All	Last Collected Oct 29, 2003 5:04:54 PM
▷Archive Area	Some	Last Collected Oct 29, 2003 4:34:21 PM
▷Data Guard	Some	Not Collected
▷Database Files	None	Last Collected Oct 29, 2003 5:17:06 PM
▷Database Job Status	All	Not Collected
▷Database Limits	Some	Last Collected Oct 29, 2003 5:11:13 PM
▷Database Services	None	Last Collected Oct 29, 2003 5:17:06 PM
▷Deferred Transactions	All	Not Collected
▷Dump Area	Some	Last Collected Oct 29, 2003 4:49:21 PM
▷Efficiency	None	Last Collected Oct 29, 2003 5:11:13 PM
▷Invalid Objects	None	Not Collected
▷Invalid Objects by Schema	All	Not Collected
▷Num of Sessions Waiting (EvtCLs)	None	Last Collected Oct 29, 2003 5:17:06 PM
▷Recovery Area	None	Not Collected
▷Response	Some	Last Collected Oct 29, 2003 5:16:51 PM
▷SGA Pool Wastage	None	Last Collected Oct 29, 2003 5:15:45 PM
▷SQL Response Time	All	Last Collected Oct 29, 2003 5:18:15 PM
▷Session Suspended	None	Not Collected
▷Snap Shot Too Old	None	Not Collected
▷System Response Time Per Call	None	Last Collected Oct 29, 2003 5:17:06 PM
▷Tablespaces Full	All	Last Collected Oct 29, 2003 5:00:46 PM
▷Tablespaces With Problem Segments	Some	Not Collected
▷Throughput	None	Last Collected Oct 29, 2003 5:11:13 PM
▷User Audit	Some	Last Collected Oct 29, 2003 4:45:45 PM

For each of these metrics, you are able to define warning and critical threshold values, and whenever the threshold is crossed, Oracle issues an alert.

Alerts are displayed on the Database Home page under the Alerts heading (or Related Alerts for non-database alerts such as a component of Oracle Net). Figure 10–4 shows five alerts including two Tablespace Full warnings.

Figure 10–4 Alerts Section of Enterprise Manager Home Page



Category All Critical 0 Warning ⚠ 5

Severity ▽	Category	Name	Message	Alert Triggered
⚠	User Audit	Audited User	User SYS logged on from stadh43.us.oracle.com.	Jun 7, 2005 2:29:58 PM
⚠	User Audit	Audited User	User SYS logged on from stadh43.	Jun 7, 2005 2:29:58 PM
⚠	Tablespaces Full	Tablespace Space Used (%)	Tablespace TVMDS_ASSM is 87 percent full	Jun 7, 2005 2:29:47 PM
⚠	Tablespaces Full	Tablespace Space Used (%)	Tablespace TVMDS_MSSM_NEW is 86 percent full	Jun 7, 2005 2:29:47 PM
⚠	Alert Log Error Status	Generic Alert Log Error Status	1 distinct types of ORA- errors have been found in the alert log.	Jun 7, 2005 2:26:17 PM

Setting thresholds is discussed in "Setting Metric Thresholds" on page 10-6. Actions you might take to respond to alerts are discussed in "Responding to Alerts" on page 10-7.

When the condition that triggered the alert is resolved and the metric's value is no longer outside the boundary, Oracle clears the alert. Metrics are important for measuring the health of the database and serve as input for self-tuning and recommendations made by Oracle advisors.

Setting Metric Thresholds

Oracle provides a set of predefined metrics, some of which initially have thresholds defined for them. There may be times when you want to set thresholds for other metrics, or you want to alter existing threshold settings.

One means of setting a threshold was introduced in "Creating a Tablespace" on page 6-8, where you set warning and critical thresholds on the amount of space consumed in a tablespace.

A more general means of setting thresholds is available using the Edit Thresholds page.

The following steps describe how to set metric thresholds:

1. From the Database Home page, click **Manage Metrics** under the Related Links heading.

   The Manage Metrics page is displayed. It displays the existing thresholds for metrics and any response actions that have been specified.

2. Click **Edit Thresholds**.

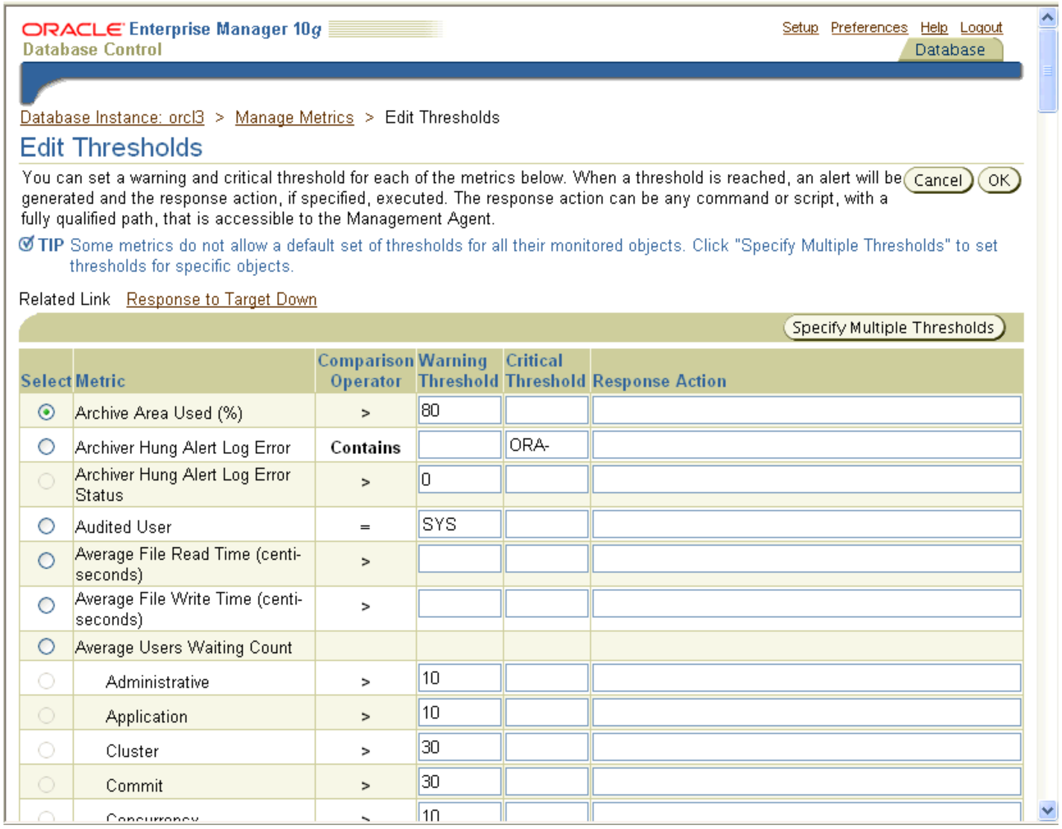
   The Edit Thresholds page, shown in Figure 10–5, is displayed. On this page you can enter new **Warning Threshold** and **Critical Threshold** values, or you can modify existing values. In the **Response Actions** field, you can enter an operating system command or a script that you want executed when an alert is issued because a threshold has been crossed.

   For example, to modify the warning threshold for **Tablespace Space Used (%)** metric, you can enter `87` as the percentage. Under **Response Actions**, you can optionally specify a fully qualified path to an operating system script that will clean up or increase the size of the tablespace.

   Oracle provides a number of alerts that are not enabled by default. You can enable them by specifying threshold values. For example, to enable the alert for **Cumulative Logons (for each second)**, enter `10` for warning and `25` for critical. This will cause the system to warn you when the number of logons each second exceeds 10.

   Click **OK** to save your changes.

Figure 10–5 Edit Thresholds Page



Database Instance: orcl3 > Manage Metrics > Edit Thresholds

Edit Thresholds

You can set a warning and critical threshold for each of the metrics below. When a threshold is reached, an alert will be generated and the response action, if specified, executed. The response action can be any command or script, with a fully qualified path, that is accessible to the Management Agent.

TIP Some metrics do not allow a default set of thresholds for all their monitored objects. Click "Specify Multiple Thresholds" to set thresholds for specific objects.

Related Link Response to Target Down

Select	Metric	Comparison Operator	Warning Threshold	Critical Threshold	Response Action
●	Archive Area Used (%)	>	80		
○	Archiver Hung Alert Log Error	Contains		ORA-	
○	Archiver Hung Alert Log Error Status	>	0		
○	Audited User	=	SYS		
○	Average File Read Time (centi-seconds)	>			
○	Average File Write Time (centi-seconds)	>			
○	Average Users Waiting Count				
○	Administrative	>	10		
○	Application	>	10		
○	Cluster	>	30		
○	Commit	>	30		
○	Concurrency	>	10		

3. For more comprehensive management of threshold settings for the different alerts, click the radio button in the **Select** column for that metric, then click **Specify Multiple Thresholds**.

   The Specify Multiple Thresholds: *metric_name* page appears. This page enables you to specify metric thresholds for each object.

   For example, to set the **Tablespace Space Used (%)** metric thresholds for individual tablespaces, select this metric then click **Specify Multiple Thresholds**. Enter the tablespace name and its warning and critical values. Click **OK**.

Responding to Alerts

When you receive an alert, follow any recommendations it provides, or consider running ADDM or another advisor, as appropriate to get more detailed diagnostics of system or object behavior.

For example, if you receive a Tablespace Space Usage alert, you might take a corrective measure by running the Segment Advisor on the tablespace to identify possible objects for shrinking. You can then shrink the objects to free space. See "Reclaiming Wasted Space" on page 6-10 in Chapter 6, "Managing Database Storage Structures".

Additionally, as a response, you can set a corrective script to run as described in "Setting Metric Thresholds" on page 10-6.

Clearing Alerts

Most alerts, such as the Out of Space alert, are cleared automatically when the cause of the problem disappears. However, other alerts such as Generic Alert Log Error are sent to you for notification and need to be acknowledged by you, the system administrator.

After taking the necessary corrective measures, you can acknowledge an alert by clearing or purging it. Clearing an alert sends the alert to the Alert History, which is viewable from the home page under Related Links. Purging an alert removes it from the Alert History.

To clear an alert such as Generic Alert Log Error, from the Home page under Diagnostic Summary, click the Alert Log link.The Alert Log Errors page appears. Select the alert to clear and click **Clear**. To purge an alert, select it and click **Purge**. You can also **Clear Every Open Alert** or **Purge Every Alert** using these buttons.

Setting up Direct Alert Notification

Enterprise Manager will display all alerts on the home page. However, you can optionally specify that Enterprise Manager provide you direct notification when specific alerts arise. For example, if you specify that you want e-mail notification for critical alerts, and you have a critical threshold set for the **system response time for each call** metric, then you might be sent an e-mail containing a message similar to the following:

```
Host Name=mydb.us.mycompany.com
Metric=Response Time per Call
Timestamp=08-NOV-2003 10:10:01 (GMT -7:00)
Severity=Critical
Message=Response time per call has exceeded the threshold. See the
lattest ADDM analysis.
Rule Name=
Rule Owner=SYSMAN
```

The e-mail contains a link to the host name and the latest ADDM analysis.

By default, alerts in critical state such as DB Down, Generic Alert Log Error Status, and Tablespace Used are set up for notification. However, to receive these notifications, you must set up your e-mail information.You can do so as follows:

1. From any Database Control page, click the **Setup** link, which is visible in the header and footer area.

2. On the Setup page, select **Notification Methods**.

3. Enter the required information into the Mail Server portion of the Notifications Methods page. See the online help for assistance.

   There are other methods of notification, including scripts and SNMP (Simplified Network Management Protocol) traps. The latter can be used to communicate with third-party applications.

   So far, you have set up a method of notification, but you have not set up an e-mail address to receive the notification. To do so, complete the following steps.

4. From any Database Control page, click the **Preferences** link, which is visible in the header and footer area.

5. On the Preferences page, select **General**. Enter your e-mail address in the **E-mail Addresses** section.

6. You can optionally edit notification rules, such as to change the severity state for receiving notification. To do so, select **Notification Rules**. The Notification Rules page appears. For more information about configuring notification rules, see *Oracle Enterprise Manager Advanced Configuration*.

Diagnosing Performance Problems

At times database performance problems arise that require your diagnosis and correction. Sometimes problems are brought to your attention by users who complain about slow performance. Other times you might notice performance spikes in the Host CPU chart on the home page.

In all cases, these problems are flagged by the Automatic Database Diagnostics Monitor (ADDM), which does a top-down system analysis every hour by default and reports its findings on the Oracle Enterprise Manager Home page.

> **See Also:** For more information about ADDM, see the *Oracle Database Performance Tuning Guide*

Viewing Performance Analysis

ADDM runs automatically every hour to coincide with the snapshots taken by Automatic Workload Repository (AWR). Its output consists of a description of each problem it has identified, and a recommended action.

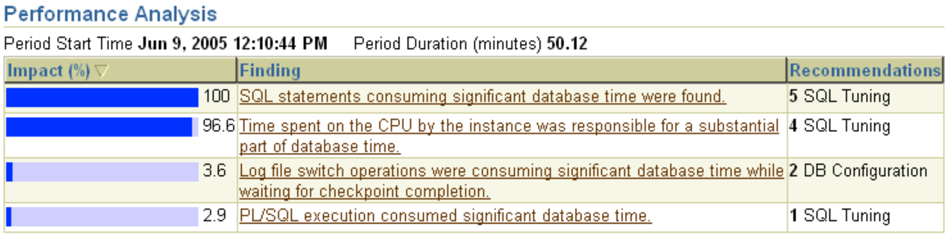
Findings are displayed in two places on the home page:

- Under the **Performance Analysis** section on the Database Home page, as shown in Figure 10–6.

  You can drill down by clicking the finding. The Performance Findings Details page appears, describing the findings and recommended actions.

- The **Diagnostic Summary** next to Performance Findings shows the number of findings, if any. Click this link to go to the ADDM page.

Figure 10–6 Performance Analysis



Performance Analysis

Period Start Time **Jun 9, 2005 12:10:44 PM** Period Duration (minutes) **50.12**

Impact (%) ▽	Finding	Recommendations
100	SQL statements consuming significant database time were found.	**5** SQL Tuning
96.6	Time spent on the CPU by the instance was responsible for a substantial part of database time.	**4** SQL Tuning
3.6	Log file switch operations were consuming significant database time while waiting for checkpoint completion.	**2** DB Configuration
2.9	PL/SQL execution consumed significant database time.	**1** SQL Tuning

Responding to Performance Findings

To respond to a performance finding, click the finding and follow the recommended actions, if any. A recommendation can include running an advisor.

For example, Figure 10–6 shows a performance finding of SQL statements consuming significant database time were found, with an impact of 100 percent and recommendation of SQL Tuning.

Clicking the finding link takes you to the Performance Finding Details page. Here the recommended action is to run the SQL Advisor, which you can do by clicking **Run Advisor Now**. The advisor runs and gives a recommendation in the form of precise actions for tuning the SQL statements for better performance.

Modifying Default ADDM Behavior

ADDM behavior and analysis is based on the Automatic Workload Repository (AWR), which collects system performance statistics and stores the data in the database. After default installation, the AWR captures data every hour in the form of a snapshot and purges data over seven days old. You can modify both the snapshot frequency and the data retention period as desired.

You can view and alter these settings on the Workload Repository page:

- The retention period for snapshots. This is initially set to 7 days. AWR automatically purges any snapshot older than the retention period, with the exception preserved snapshots that are retained for ever. A preserved snapshot set is typically created to define a reference period for performance analysis.
- The interval for snapshots. The default and recommended value is one hour.

To navigate to this page, from the Database Administration page, under Statistics Management, select **Automatic Workload Repository**.

To change either of these settings, click **Edit** on the Workload Repository page. The Edit Settings page appears. Enter a new Snapshot retention period or new System Snapshot Interval. Click **OK**.

Running ADDM Manually

By default Oracle runs ADDM every hour. Performance findings, if any, from the last snapshot are listed on the Oracle Enterprise Manager Home page. This is described in "Performance Self-Diagnostics: Automatic Database Diagnostics Monitor" on page 10-2.

You can also run ADDM manually. Reasons for doing so include running it as a recommended action associated with an alert, running it in the middle of a snapshot period, or running it across multiple snapshots.

Note: If you need more frequent ADDM reporting, you can also modify the default snapshot interval. To do so, see "Modifying Default ADDM Behavior" on page 10-10.

From the Home page, under Related links you can navigate to the ADDM page by clicking **Advisor Central**, then **ADDM**. The Run ADDM page appears.

Figure 10–7 is a screen shot of Run ADDM page.

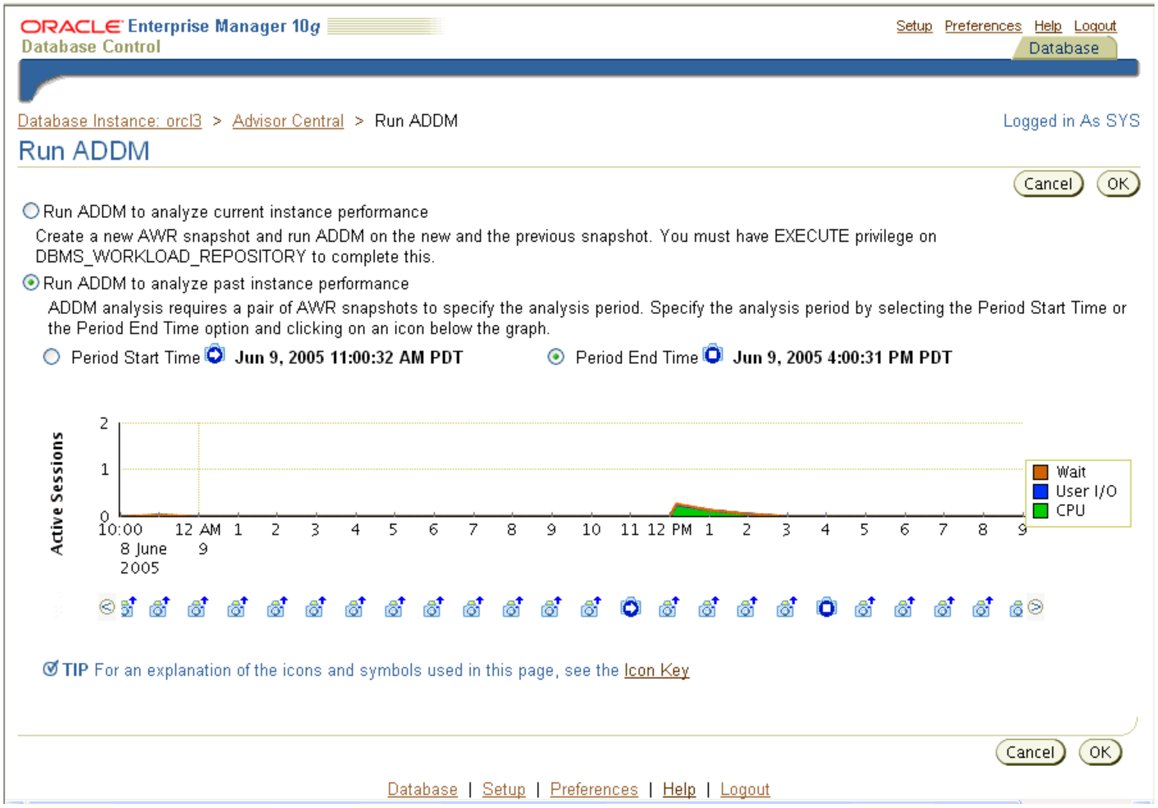
Increased session activity is displayed as peaks in the graph.

To analyze a period across multiple snapshots:

1. Select **Run ADDM to analyses past instance performance**.
2. Select **Period Start Time**. Choose a starting snapshot under the graph.
3. By default, the end time is the last snapshot. To specify a different end time, select **Period End Time**. Choose an ending snapshot under the graph.

4. Click **OK** to start the analysis. The Automatic Database Diagnostic Monitor page appears detailing its findings and recommendations.

Figure 10–7 Run ADDM Page



Database Instance: orcl3 > Advisor Central > Run ADDM

Logged in As SYS

Run ADDM

○ Run ADDM to analyze current instance performance

Create a new AWR snapshot and run ADDM on the new and the previous snapshot. You must have EXECUTE privilege on DBMS_WORKLOAD_REPOSITORY to complete this.

● Run ADDM to analyze past instance performance

ADDM analysis requires a pair of AWR snapshots to specify the analysis period. Specify the analysis period by selecting the Period Start Time or the Period End Time option and clicking on an icon below the graph.

○ Period Start Time **Jun 9, 2005 11:00:32 AM PDT** ● Period End Time **Jun 9, 2005 4:00:31 PM PDT**

TIP For an explanation of the icons and symbols used in this page, see the Icon Key

Using Advisors to Optimize Database Performance

Advisors are powerful tools for database management. They provide specific advice on how to address key database management challenges, covering a wide range of areas including space, performance, and undo management. In general, advisors produce more comprehensive recommendations than alerts. This is because alert generation is intended to be low cost and have minimal impact on performance, whereas advisors are user-invoked, consume more resources, and perform more detailed analysis. This, along with the what-if capability of some advisors, provides vital information for tuning that cannot be procured from any other source.

About Performance Advisors

This section deals primarily with the advisors that can improve performance. These advisors include the SQL Tuning, SQL Access, and Memory Advisors. Table 10–1, " Performance Advisors" describes these advisors.

Other advisors such as the Undo and Segment Advisors are listed in Table 10–2, " Other Advisors".

For example, the shared pool memory advisor graphically displays the impact on performance of changing the size of this component of the SGA.

You can run an advisor when faced with the following situations:

- You want to resolve a problem in a specific area, for example, to determine why a given SQL statement is consuming 50 percent of CPU time and what to do to reduce its resource consumption.You can use the SQL Tuning Advisor.
- During application development, you want to tune new schema and its associated SQL workload for optimal performance. You can use the SQL Access Advisor.
- You are planning to add memory to your system. You can use the Memory Advisor to determine the database performance impact of increasing your SGA or PGA.

Table 10–1 describes the performance advisors that Oracle provides. These advisors are described in this chapter.

Table 10–1 Performance Advisors

Advisor	Description
Automatic Database Diagnostic Monitor (ADDM)	ADDM makes it possible for the Oracle Database to diagnose its own performance and determine how any identified problems can be resolved. See "Performance Self-Diagnostics: Automatic Database Diagnostics Monitor" and "Diagnosing Performance Problems" on page 10-9.
SQL Tuning Advisor	This advisor analyzes SQL statements and makes recommendations for improving performance. See "Using the SQL Tuning Advisor" on page 10-13.
SQL Access Advisor	Use this advisor to tune schema to a given SQL workload. For example, the SQL Access Advisor can provide recommendations for creating indexes and materialized views for a given workload. See "Using the SQL Access Advisor" on page 10-14.
Memory Advisor ■ Shared Pool Advisor (SGA) ■ Buffer Cache Advisor (SGA) ■ PGA Advisor	By default, Oracle automatically tunes physical memory allocation for optimal performance. The Memory Advisor gives graphical analysis of SGA and PGA settings, which you can use for what-if planning. See "Using the Memory Advisor" on page 10-15.

Table 10–2, " Other Advisors" describes other advisors Oracle provides. These are described elsewhere in this book.

Table 10–2 Other Advisors

Advisor	Description
Segment Advisor	The Segment Advisor provides advice on whether an object is a good candidate for a shrink operation based on the level of space fragmentation within that object. The advisor also reports on the historical growth trend of segments. You can use this information for capacity planning and for arriving at an informed decision about which segments to shrink. See "Reclaiming Wasted Space" in Chapter 6, "Managing Database Storage Structures".
Undo Advisor	The Undo Advisor helps identify problems in the undo tablespace and assists in correctly sizing the undo tablespace. The Undo Advisor can also be used to set the low threshold value of the undo retention period for any flashback requirements. See "Using the Undo Advisor" in Chapter 6, "Managing Database Storage Structures".

You can run an advisor from the Advisor Central home page, accessible through a link on the Database Home page. You can invoke advisors in other ways, often through recommendations from ADDM or alerts.

Using the SQL Tuning Advisor

Use the SQL Tuning Advisor for tuning SQL statements. Typically, you can run this advisor in response to an ADDM performance finding that recommends its use.

Additionally, you can run the SQL Tuning Advisor on the most resource-intensive SQL statements, as well as on a SQL workload.

> **See Also:** For more information about the SQL Tuning advisor, see *Oracle Database Performance Tuning Guide*

To run the SQL Tuning Advisor do the following:

1. On the Home Page, under Related Links, click **Advisor Central**, then click **SQL Tuning Advisor**. The SQL Tuning Advisor Links page appears.
2. The advisor can be run on one of the following sources:
   - Top Activity—The most resource intensive SQL statements executed during the last hour. Use this option to tune SQL statements that might have caused recent performance problems.
   - Period SQL— A set of SQL statements over any 24 hour window. Use this option for pro-active tuning of SQL statements.
   - SQL Tuning Sets (STS)—A set of SQL statements you provide. An STS can be created from SQL statements captured by AWR snapshots or from any SQL workload.
   - Snapshots—A set of SQL statements from an AWR snapshot.
   - Preserved Snapshot Sets—A set of SQL statements from a preserved snapshot set.
3. For example, select **Top Activity**.

   The Top Activity page appears. This page has a graph showing the active sessions over the last hour.You can select an five minute interval to analyze by clicking the bar under the time line. The Top SQL and Top Sessions tables show the activity within the selected period. You can select one or more SQL statements to analyze.
4. Click **Schedule SQL Tuning Advisor**.

   The Schedule Advisor page appears showing the SQL statements in the interval. Give your task a name and description, select the scope for the analysis **(Comprehensive or Limited)**, and select a start time for the task. Click **OK**.
5. The SQL Tuning Result page appears. To view recommendations, select the SQL statement and click **View Recommendations**. The recommendation can include one or more of the following:
   - Accept the SQL profile, which contains additional SQL statistics specific to the statement that enables the query optimizer to generate a significantly better execution plan at runtime.
   - Gather optimizer statistics on objects with outdated or no statistics
   - Advice on how to rewrite a query for better performance.
   - Create an index to offer alternate, faster access paths to the query optimizer.

Using the SQL Access Advisor

The SQL Access Advisor helps define appropriate access structures such as indexes and materialized views to optimize SQL queries. The advisor takes a SQL workload as an input and recommends which indexes, materialized views, or logs to create, drop, or retain for faster performance. You can select your workload from different sources including current and recent SQL activity, a SQL repository, or a user-defined workload such as from a development environment.

The recommendations that this advisor makes include possible indexes, materialized views, or materialized view logs that can improve your query performance for the given workload.

Note that both SQL Tuning and Access advisor provide index creation recommendations. The SQL Tuning advisor recommends creation of indexes only when it anticipates exceptional performance gains for the SQL statement being tuned. However, creation of new indexes may adversely impact the performance of DML (inserts, updates, and deletes) operations. The SQL Tuning advisor does not take this into account while generating new index recommendations.

The SQL Access Advisor on the other hand considers the impact of new indexes on the complete workload. As such, if an index improves performance of one SQL statement but adversely impacts the rest of the workload, the new index won't be recommended by the SQL Access advisor. For this reason, the SQL Tuning advisor always recommends validating its new index recommendation by running the SQL Access Advisor.

> **See Also:** For more information about the SQL Access advisor, see *Oracle Database Performance Tuning Guide*

To run the SQL Access advisor:

1. Navigate to the Advisor Central page, and click **SQL Access Advisor**. This begins a wizard which prompts you for your initial options. Select Use Default Options. Click **Continue**.
2. The Workload Source page appears. Select your workload source as **Current and Recent SQL Activity** to analyze the most recent and ongoing activity. You can also select from other sources. The best workload is one that fully represents the statements that access all the underlying tables. Click **Next.**
3. The Recommendations Options page appears. Select if you want the advisor to recommend indexes, materialized views, or both. You can also choose to perform an evaluation only, which returns information on the objects being accessed by the workload, but does not recommend any new structures.

   You can choose to run the advisor in limited or comprehensive mode. Limited mode runs faster by concentrating on highest cost statements. You can ignore the Advanced Options for now. Click **Next.**
4. The Schedule page appears. Ensure your task has a name and description. Under Start, select Immediately. Click **Next.**
5. The Review page appears. Review your options and click **Submit**.
6. The Advisor Central page appears. A confirmation message appears, indicating that your task has been created successfully. Click **Refresh** to view the status of your task.
7. When your SQL Access Advisor task has completed, select **View Result**. The Result for Task page appears.

The Summary page shows you the potential for improvement under the headings Workload I/O Cost and Query Execution Time Improvement.

The Recommendations page shows the recommendations, if any, for improving performance. A recommendation might consist of a SQL script with one or more `CREATE INDEX` statements, which you can run by clicking **Schedule Implementation**.

Using the Memory Advisor

Adequate physical memory has a significant impact on the performance of your Oracle Database. The SGA and PGA target initialization parameters determine the amount of physical memory available to the database. When you use the DBCA to create your database, the initial value of these parameters is configured according to your workload type and the total amount of memory available on your machine.

With Oracle's automatic memory management capabilities, Oracle automatically adjusts the memory distribution among the various SGA and PGA sub-components for optimal performance. These adjustments are made within the boundaries of your total SGA and PGA target values.

ADDM periodically evaluates the performance of your database to determine performance bottlenecks. If ADDM finds that the current amount of available memory is inadequate and adversely affecting performance, it can recommend that you increase your SGA or PGA target value. You can set new values for the SGA and PGA using the Memory Advisor.

Additionally, you can use the Memory Advisor to perform what-if analysis on:

- The database performance benefits of adding physical memory to your database.
- The database performance impact of reducing the physical memory available to your database.

Optionally, you can use the Memory Advisor to set a new value for the SGA and PGA targets, based on what-if analysis.

> **See Also:** For more information about memory configuration, see *Oracle Database Performance Tuning Guide*

Setting New SGA or PGA Targets

As a response to an ADDM performance finding, you can use the Memory Advisor to set a new SGA or PGA target. In the user-interface, these parameters correspond to **Total SGA Size** and **Aggregate PGA Target**.

To set a new SGA target:

1. Navigate to the memory advisor in one of the following ways:

   - From the Home Page, under related links, click **Advisor Central**, then Click **Memory Advisor**.
   - From the Administration page, under Database Configuration, click **Memory Parameters**.

     The Memory Parameters: SGA page appears. This page gives the breakdown of memory allocation for the system global area (SGA) and its subcomponents such as the buffer cache and shared pool. For more information about these components, see "Instance Memory Structure" in Chapter 5, "Managing the Oracle Instance".

2. To set a new SGA size, enter the new amount for Total SGA Size. The new value must be less than the maximum SGA. Click **Apply**. A confirmation message appears.

Similarly, to set a new PGA size:

1. Navigate to the PGA property page. This page shows the current value for the Aggregate PGA Target, as well as the current allocation of PGA memory. For more information about the PGA, see "Instance Memory Structure" in Chapter 5, "Managing the Oracle Instance".
2. Enter the new amount next to **Aggregate PGA Target**. Click **Apply**. A confirmation message appears.

Determining Performance Impact for Altering your SGA or PGA

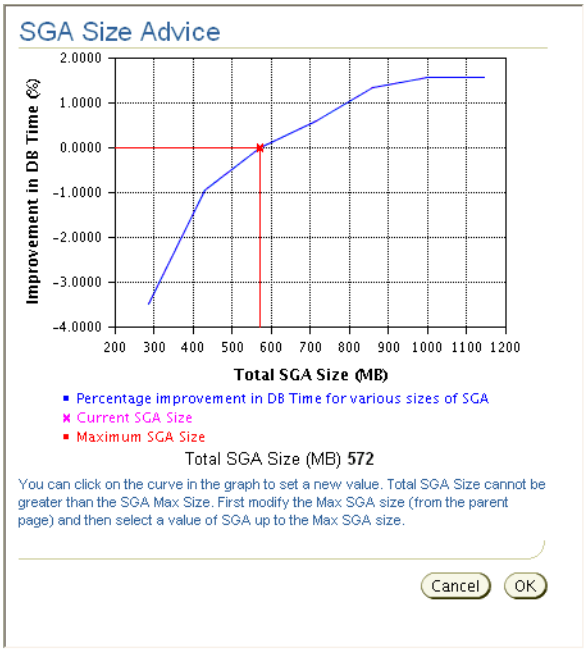
You can use the Memory Advisor to do what-if analysis for adding or removing physical memory to your database. This advisor graphically analyses the database performance impact of altering your SGA or PGA targets.

To gain advice on configuring the total size of your SGA, on the Memory Parameters: SGA Page, click **Advice** next to Total SGA Size.

The SGA Advice graph appears. Refer to Figure 10–8, "SGA Size Advice".

Improvement in DB Time (%) is plotted against Total SGA size. A higher number for Improvement in DB Time is better for performance. In this example, the graph tells us that an SGA size larger than 1000MB will not improve performance by much. Thus 1000MB is the recommended optimal SGA size.

Figure 10–8 SGA Size Advice



Total SGA Size (MB) **572**

You can click on the curve in the graph to set a new value. Total SGA Size cannot be greater than the SGA Max Size. First modify the Max SGA size (from the parent page) and then select a value of SGA up to the Max SGA size.

Similarly, to run the PGA advisor, navigate to the **PGA** property page. Next to Aggregate PGA Target, click **Advice**. The PGA Target Advice Page appears, plotting Cache hit percentage against PGA Target memory size.

Cache hit percentage is the percentage of read requests serviced by memory, as opposed to those requests serviced by reading from disk, which is slower. A higher hit percentage indicates better cache performance. The optimal zone for cache hit percentage is between 75 and 100 percent. However, it is not safe to conclude that your database is having performance problems simply because your cache hit percentage is not within the optimal zone. When the amount of currently available PGA memory is not adequate for optimal performance, ADDM will automatically recommend adjusting this value with a performance finding.

Monitoring and Tuning: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter, and includes annotated screen shots.

To view the Monitoring OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/monitoring/monitoring.htm
```

11

Managing Oracle Software

This chapter describes how to keep your Oracle software up-to-date with patch releases. This chapter contains the following topics:

- About Software Management and Patch Releases
- Applying a Patch Release

About Software Management and Patch Releases

Software management involves keeping your Oracle software up-to-date with the latest product fixes.

Periodically, Oracle issues product fixes for its software called patch releases. Patch releases are fully tested product fixes only. They do not include new functionality. Application of a patch release affects the software residing in your Oracle home only, with no upgrade or change to the database.

Patches are individual bug fixes. Patch sets are a collection of bug fixes up to the time of the patch set release.

All patch and patch set releases carry version numbers. For example, if you bought Oracle 10*g* Release 10.2.0.1, then an available patch set might be 10.2.0.2. Every patch or patch set also has a patch number to identify it.

Every patch release has an associated README file that describes its bug fixes. The README also has instructions for manually applying the patch.

Enterprise Manager enables you to find the latest patch release on the Oracle MetaLink Web site and to download it to your Oracle home.

Applying a Patch Release

The steps involved in applying a patch are as follows:

- Determining your Oracle Software Environment
- Setting Your Oracle MetaLink Credentials
- Staging the Patch Release
- Installing the Patch Release

Determining your Oracle Software Environment

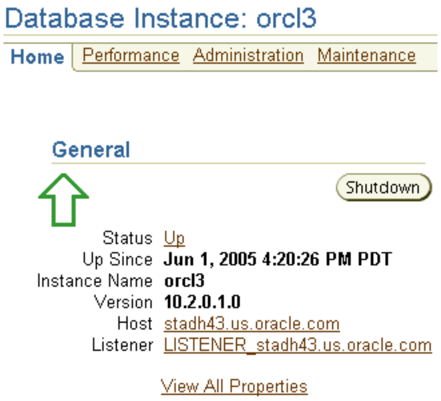
To select the appropriate patch for your environment, you need to know:

- What version of the Oracle database you are running

- The location of your Oracle home.
- What type of hardware configuration you have

You can find this information in the General section of the Database Home page, as shown in Figure 11–1.

Figure 11–1 General Section of Database Home Page



Database Instance: orcl3

Status Up
Up Since **Jun 1, 2005 4:20:26 PM PDT**
Instance Name **orcl3**
Version **10.2.0.1.0**
Host stadh43.us.oracle.com
Listener LISTENER_stadh43.us.oracle.com

Determining the Oracle Database Version

You can determine your Oracle database version by navigating to the General section of the Enterprise Manager home page. For example, Figure 11–1 shows that the database version is 10.2.0.1.0.

Determining the Location of your Oracle Home

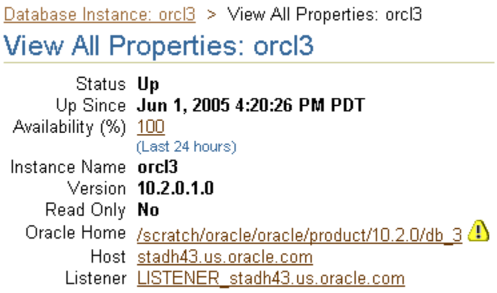
The Oracle home is the operating system location of your Oracle installation.

To determine the Oracle home:

1. In the General section of Enterprise Manager home page, click **View All Properties**.

   The View All Properties: *instance_name* page appears. As shown in Figure 11–2, this page displays the path for the Oracle home.

Figure 11–2 View All Properties Page



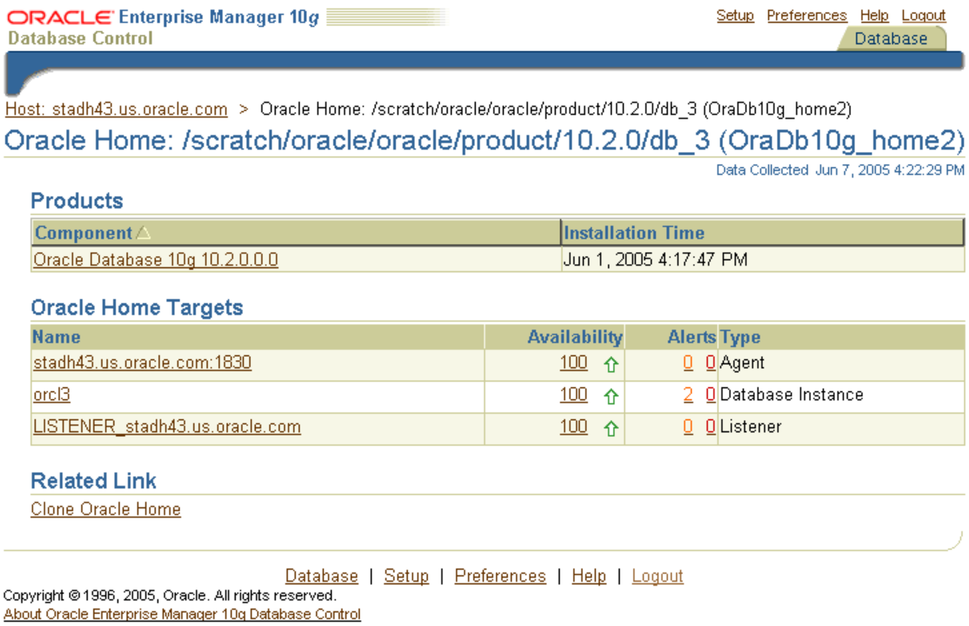
View All Properties: orcl3

Status **Up**
Up Since **Jun 1, 2005 4:20:26 PM PDT**
Availability (%) 100
(Last 24 hours)
Instance Name **orcl3**
Version **10.2.0.1.0**
Read Only **No**
Oracle Home /scratch/oracle/oracle/product/10.2.0/db_3
Host stadh43.us.oracle.com
Listener LISTENER_stadh43.us.oracle.com

2. Click the path link next to **Oracle Home**.

   The Oracle home: *oracle_home* page appears. As shown in Figure 11–3, this page displays the file directory of your Oracle home, along with the installed product components and time of installation.

Figure 11–3 Oracle Home Page



Host: stadh43.us.oracle.com > Oracle Home: /scratch/oracle/oracle/product/10.2.0/db_3 (OraDb10g_home2)

Oracle Home: /scratch/oracle/oracle/product/10.2.0/db_3 (OraDb10g_home2)

Data Collected Jun 7, 2005 4:22:29 PM

Products

Component △	Installation Time
Oracle Database 10g 10.2.0.0.0	Jun 1, 2005 4:17:47 PM

Oracle Home Targets

Name	Availability	Alerts		Type
stadh43.us.oracle.com:1830	100 ⇧	0	0	Agent
orcl3	100 ⇧	2	0	Database Instance
LISTENER_stadh43.us.oracle.com	100 ⇧	0	0	Listener

Related Link

Clone Oracle Home

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2005, Oracle. All rights reserved.

About Oracle Enterprise Manager 10g Database Control

3. In the Component section, click the link to drill down to the page that describes specific components in your installation, including all component dependencies. In the Oracle Home Target section, click the links to display pages for the host agent, instance, and listener.

Determining Your Hardware Configuration

You hardware configuration consists of your database machine and operating system. You need this information to select the appropriate Oracle patch.

To learn more about your hardware configuration:

1. In the General section of Enterprise Manager home page, click the link next to **Host**.

   The Host: *host_name* page appears. This page lists your operating system, hardware platform, IP address, number of CPUs, memory size, and disk space.

2. In the Configuration section, click the links to drill down to pages that describe your operating system, hardware platform, and disk space.

Setting Your Oracle MetaLink Credentials

You obtain patches and patch releases from Oracle MetaLink, which is the Oracle support Web site:

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

To download patches from Metalink, you need to register using your customer support identifier, which your company obtains when it signs a support contract with Oracle. On registering with MetaLink, you are given a username and password that enables you to log in to this site from a Web browser.

Alternatively, you can give Enterprise Manager your login credentials so that it can log in to MetaLink automatically and search for patch releases. You must set these credentials before you can run the patch wizard in Enterprise Manager. The patch wizard can automatically search and download patches for you, and is described in section "Staging the Patch Release" on page 11-4.

To set your MetaLink login credentials:

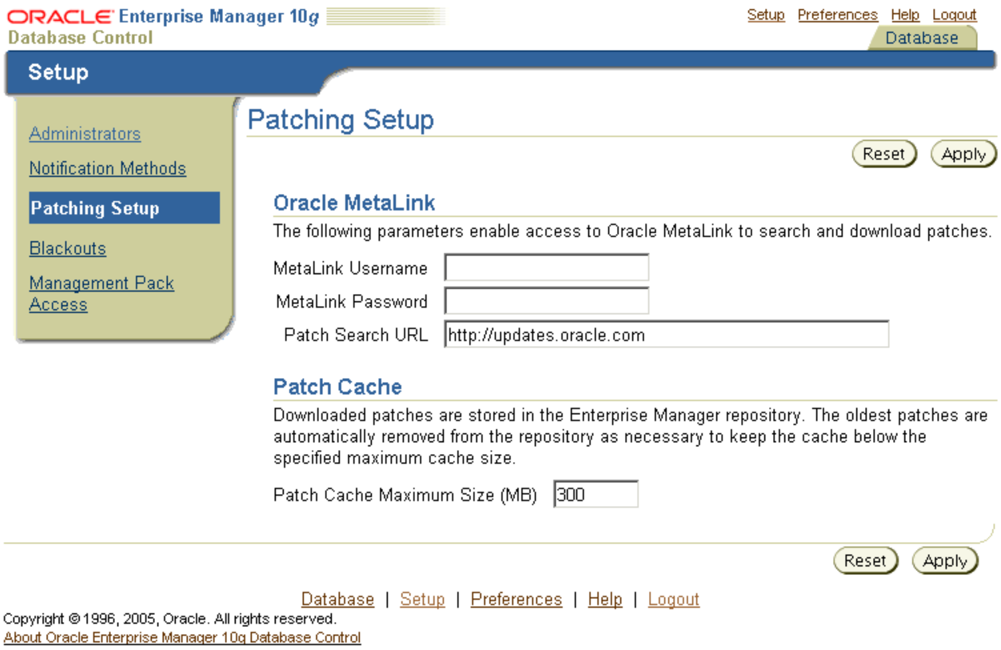
1. Click the **Setup** link located at the top and bottom of most Enterprise Manager pages.

   The Setup page appears.

2. Click **Patching Setup**.

   The Patching Setup page appears, as shown in Figure 11–4.

Figure 11–4 Patching Setup Page



Patching Setup

Oracle MetaLink

The following parameters enable access to Oracle MetaLink to search and download patches.

MetaLink Username

MetaLink Password

Patch Search URL http://updates.oracle.com

Patch Cache

Downloaded patches are stored in the Enterprise Manager repository. The oldest patches are automatically removed from the repository as necessary to keep the cache below the specified maximum cache size.

Patch Cache Maximum Size (MB) 300

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2005, Oracle. All rights reserved.

About Oracle Enterprise Manager 10g Database Control

3. In the **MetaLink Username** and **MetaLink Password** boxes, enter the username and password that you obtain by registering with your customer support id. In **Patch Cache Maximum Size (MB)**, you can optionally set the maximum size of your patch cache, a temporary area where patches are stored. The oldest patches are automatically removed to keep the patch cache under its maximum size.

4. Click **Apply** to set your MetaLink credentials.

Staging the Patch Release

You can view available patch releases at MetaLink by using Enterprise Manager. Viewing these updates is the first step in the patching wizard, which you can use to download the patch to your Oracle home.

Downloading the patch is known as staging. After staging, you can apply the patch manually by shutting down the database and running the Oracle Universal Installer.

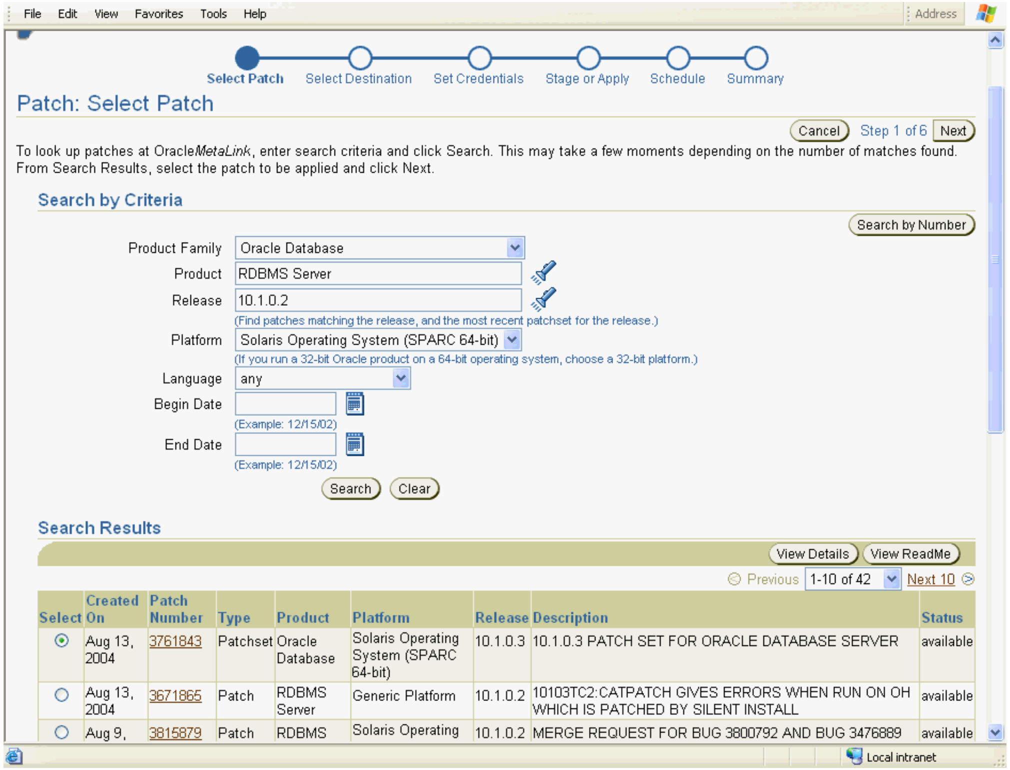
Note: The following example searches for and stages a patch release on an Oracle Database, Release 10.1.0.2. The Enterprise Manager wizard and screenshot might be slightly different for your Oracle Database, Release 10.2.

To start the patching wizard:

1. From the home page, click **Maintenance**.
2. In the Database Software Patching section, click **Apply Patch**.

   The Patch: Select Patch page appears, as shown in Figure 11–5.

Figure 11–5 Patch: Select Patch page



Patch: Select Patch

To look up patches at Oracle*MetaLink*, enter search criteria and click Search. This may take a few moments depending on the number of matches found. From Search Results, select the patch to be applied and click Next.

Search by Criteria

Product Family	Oracle Database
Product	RDBMS Server
Release	10.1.0.2 (Find patches matching the release, and the most recent patchset for the release.)
Platform	Solaris Operating System (SPARC 64-bit) (If you run a 32-bit Oracle product on a 64-bit operating system, choose a 32-bit platform.)
Language	any
Begin Date	(Example: 12/15/02)
End Date	(Example: 12/15/02)

Search Results

Select	Created On	Patch Number	Type	Product	Platform	Release	Description	Status
●	Aug 13, 2004	3761843	Patchset	Oracle Database	Solaris Operating System (SPARC 64-bit)	10.1.0.3	10.1.0.3 PATCH SET FOR ORACLE DATABASE SERVER	available
○	Aug 13, 2004	3671865	Patch	RDBMS Server	Generic Platform	10.1.0.2	10103TC2:CATPATCH GIVES ERRORS WHEN RUN ON OH WHICH IS PATCHED BY SILENT INSTALL	available
○	Aug 9,	3815879	Patch	RDBMS	Solaris Operating	10.1.0.2	MERGE REQUEST FOR BUG 3800792 AND BUG 3476889	available

3. Enterprise Manager automatically fills in values for **Product Family**, **Product**, **Release**, and **Platform**. The search results that match the criteria are displayed by most recent patch (or patch set) at the top of the list.

   Note that you can limit your search by entering values for **Begin Date** and **End Date** and then clicking **Search**.
4. Select a patch and click **View Details** to view the patch details. Select a patch and click **View ReadMe** to view the ReadMe, which includes a description of the bug fixes. Return to the Patch wizard by clicking the Patch locator link on the View Patch Details page.
5. Select the patch you are interested in, or select the most recent patch set if you are doing a periodic software update. In this example, we select patch set number 3761843, dated Aug 13, 2004, release 10.1.0.3. Click **Next**.

   The Patch: Select Destination page appears. On this page you select your destination.
6. In **Destination Type**, specify your target database or a host or directory location to stage your patch. In this example, you accept the default database target, listed under Selected Targets. Click **Next**.

   The Patch: Set Credentials page appears.

7. In **Username** and **Password** boxes, enter the operating system username and password to enable Enterprise Manager to stage the patch to your Oracle home. Enterprise Manager requires these credentials for job scheduling. Click **Next**.

   The Patch: Stage or Apply page appears.

8. By default Enterprise Manager downloads the patch to the stage directory that is listed in the main box. Typically, this location is an Oracle home subdirectory called `EMStagedPatches`*`patchnumber`*.

   By default, Enterprise Manager stages the patch only. You can then manually apply the patch by following the directions given in the README. The directions include shutting down the database and your applications and then running the Oracle Universal Installer.

   You can optionally have Oracle run an operating system script to apply the patch automatically. To do so, you check the option **Run Script to Apply Patch**, which runs the script template listed in the window. Note that if you choose this option, you must edit the template to have it shut down the database and your application processes. Oracle does not recommend this option unless you have significant experience in shell scripting.

9. Click **Next**.

   The Patch: Schedule page appears.

10. In the Schedule Patch Job section, select **Immediately** to download and stage the patch immediately. Click **Next**.

    The Patch: Summary page appears.

11. Click **Finish**.

    A confirmation page confirms that your job was submitted.

12. Confirm that the patch was staged by viewing the patch cache. Navigate to the Maintenance page, and in the **Database Software Patching** section, click **View Patch Cache**.

    The Patch Cache appears. The downloaded patch should appear in the table.

Installing the Patch Release

After you have downloaded the latest patch or patch set release, you can install it. Before you install it, you must shut down your database instances, listeners, and dependent applications, as well as back up your current Oracle home installation.

Because of these preliminary steps, Oracle recommends applying patches during off-peak hours, when the system is not actively being used. It is also helpful to notify database users in advance of any planned downtime.

When your system is down as such, you can apply the patch by unzipping the downloaded file and running the Oracle Universal Installer.

See Also: The patchset README for further instructions on installing the patch

This section describes the basic steps to take after downloading the patch release:

1. Read the README included with the patch release for patch-specific installation instructions.

2. Shut down all dependent applications.
3. Shut down your database instances. See "Shutting Down and Restarting the Instance and Database with Enterprise Manager" in Chapter 5, "Managing the Oracle Instance".
4. Optionally, shut down the database listener. This step is optional because your database listener automatically reconnects to the database when you restart the database. However, keeping the listener up can enable a client to log in to the database during the upgrade process, which might be undesirable. See "Starting and Shutting Down the Listener" in Chapter 4, "Configuring the Network Environment".
5. Shut down Enterprise Manager. See "Starting and Stopping the Oracle Enterprise Manager Console" in Chapter 3, "Getting Started with Oracle Enterprise Manager".
6. Apply the patch by following the instructions in the README.
7. Start up the database.
8. Start up the listener if you took it down previously. If you left the listener running, then it will automatically reconnect to the database. See "Starting and Shutting Down the Listener" in Chapter 4, "Configuring the Network Environment".
9. Start up Enterprise Manager. See "Starting and Stopping the Oracle Enterprise Manager Console" in Chapter 3, "Getting Started with Oracle Enterprise Manager".
10. Restart up your applications.

Managing Oracle Software: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this chapter, and includes annotated screen shots.

To view the Managing Oracle Software OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/managing/managingsoftware.htm
```

A

Automatic Storage Management

This appendix discusses using Oracle Automatic Storage Management (ASM) to provision and manage storage for the Oracle Database datafiles. It includes the following topics:

- What is Automatic Storage Management?
- Overview of Disks, Disk Groups, and Failure Groups
- Installing Automatic Storage Management
- Accessing the Automatic Storage Management Home Page
- Starting Up and Shutting Down the ASM Instance
- Monitoring Disk Group Space Usage
- Creating a Disk Group
- Dropping a Disk Group
- Adding Disks to a Disk Group
- Dropping Disks from a Disk Group
- Backing Up ASM-Managed Files

What is Automatic Storage Management?

Automatic Storage Management (ASM) is an integrated, high-performance database file system and disk manager. ASM is based on the principle that the database should manage storage instead of requiring an administrator to do it. ASM eliminates the need for you to directly manage potentially thousands of Oracle database files.

ASM groups the disks in your storage system into one or more **disk groups**. You manage a small set of disk groups and ASM automates the placement of the database files within those disk groups.

ASM provides the following benefits:

- **Striping—**ASM spreads data evenly across all disks in a disk group to optimize performance and utilization. This even distribution of database files eliminates the need for regular monitoring and I/O performance tuning.

  For example, if there are six disks in a disk group, pieces of each ASM file are written to all six disks. These pieces come in 1 MB chunks known as **extents**. When a database file is created, it is **striped** (divided into extents and distributed) across the six disks, and allocated disk space on all six disks grows evenly. When reading the file, file extents are read from all six disks in parallel, greatly increasing performance.

- **Mirroring—**ASM can increase availability by optionally mirroring any file. ASM mirrors at the file level, unlike operating system mirroring, which mirrors at the disk level. Mirroring means keeping redundant copies, or **mirrored copies**, of each extent of the file, to help avoid data loss caused by disk failures. The mirrored copy of each file extent is always kept on a different disk from the original copy. If a disk fails, ASM can continue to access affected files by accessing mirrored copies on the surviving disks in the disk group.

  ASM supports 2-way mirroring, where each file extent gets one mirrored copy, and 3-way mirroring, where each file extent gets two mirrored copies.

- **Online storage reconfiguration and dynamic rebalancing—**ASM permits you to add or remove disks from your disk storage system while the database is operating. When you add a disk, ASM automatically redistributes the data so that it is evenly spread across all disks in the disk group, including the new disk. This redistribution is known as rebalancing. It is done in the background and with minimal impact to database performance. When you request to remove a disk, ASM first rebalances by evenly relocating all file extents from the disk being removed to the other disks in the disk group.

- **Managed file creation and deletion—**ASM further reduces administration tasks by enabling files stored in ASM disk groups to be **Oracle-managed files**. ASM automatically assigns filenames when files are created, and automatically deletes files when they are no longer needed. For information about Oracle-managed files, see *Oracle Database Administrator's Guide*.

Oracle recommends that you use ASM for your database file storage, instead of raw devices or the operating system file system. However, databases can have a mixture of ASM files and non-ASM files. Oracle Enterprise Manager includes a wizard that enables you to migrate non-ASM database files to ASM.

The ASM Instance

ASM is implemented as a special kind of Oracle instance, with its own System Global Area and background processes. The ASM instance is tightly integrated with the database instance. Every server running one or more database instances that use ASM for storage has an ASM instance. In a Real Application Clusters environment, there is one ASM instance for each node, and the ASM instances communicate with each other on a peer-to-peer basis. Only one ASM instance is required for each node regardless of the number of database instances on the node.

Administering ASM

You administer ASM with Oracle Enterprise Manager Database Control or Grid Control. This Appendix assumes that you are using Database Control.

> **Note:** Refer to *Oracle Database Administrator's Guide* for additional information about ASM.

Overview of Disks, Disk Groups, and Failure Groups

You configure ASM by creating disk groups that become the default location for files created in the database. The disk group type determines how ASM mirrors files. When you create a disk group, you indicate whether the disk group is a **normal redundancy** disk group (2-way mirroring for most files by default), a **high redundancy** disk group (3-way mirroring), or an **external redundancy** disk group (no mirroring by ASM). You use an external redundancy disk group if your storage system already does mirroring at the hardware level, or if you have no need for redundant data. The default disk group type is normal redundancy.

A disk group consists of a grouping of disks that are managed together as a unit. These disks are referred to as **ASM disks**. An ASM disk can be a disk device, a partition, or a network-attached file.

When an ASM instance starts, it automatically **discovers** all available ASM disks. Discovery is the process of finding all disks that have been prepared for ASM by your system administrator, examining their disk headers, and determining which disks belong to disk groups and which are available for assignment to disk groups. ASM discovers disks in the paths that are listed in an initialization parameter, or if the parameter is `NULL`, in an operating system–dependent default path.

Failure groups define ASM disks that share a common potential failure mechanism. An example of a failure group is a set of SCSI disks sharing the same SCSI controller. Failure groups are used to determine which ASM disks to use for storing redundant copies of data. For example, if 2-way mirroring is specified for a file, ASM automatically stores redundant copies of file extents in separate failure groups. Failure groups apply only to normal and high redundancy disk groups. You define the failure groups in a disk group when you create or alter the disk group.

Note: By default, when you create a disk group, every disk in the disk group belongs to exactly one failure group. For most systems, the default failure groups work well to prevent data loss.

See Also: *Oracle Database Administrator's Guide* for additional information about failure groups, and for instructions on how to configure and use failure groups.

Installing Automatic Storage Management

This appendix assumes that Automatic Storage Management (ASM) is already installed and configured on your single-instance server or on your Real Application Clusters (RAC) nodes. For information on installing and initially configuring ASM, see the following Oracle publications:

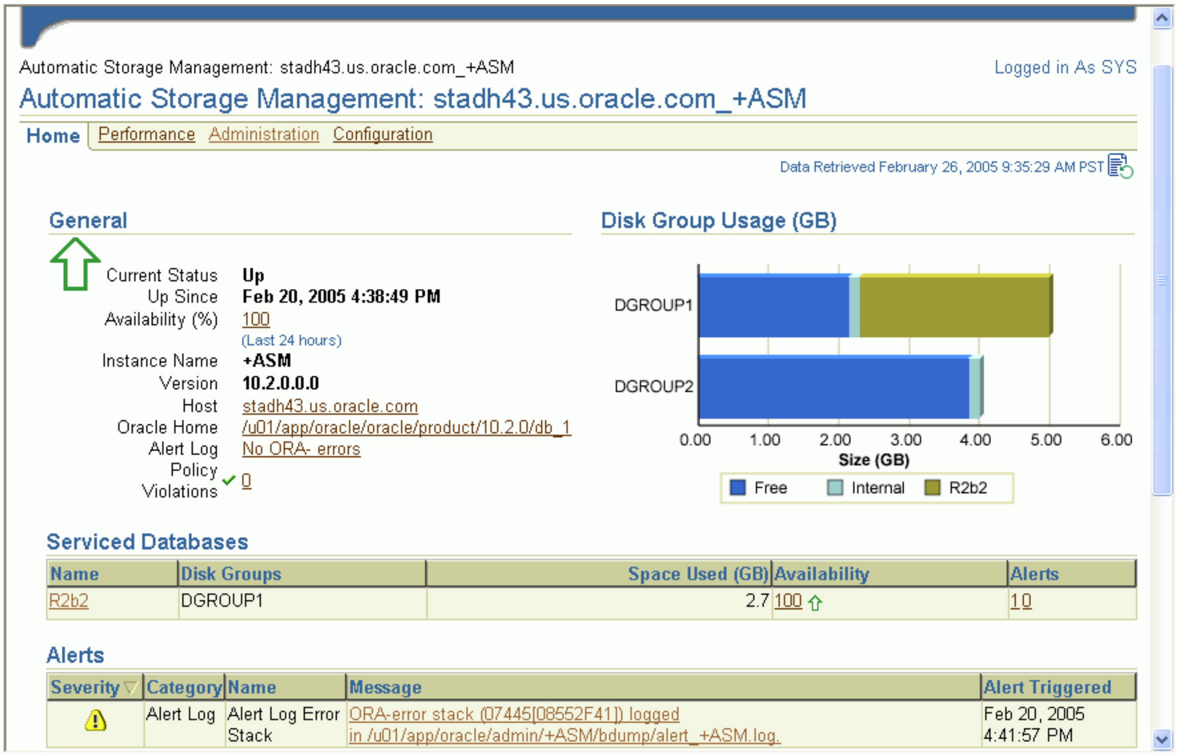
- For a single-instance server, the Installation Guide for your operating system
- For RAC, the Real Application Clusters Installation and Configuration Guide for your operating system

In addition, see *Oracle Database Administrator's Guide* for a list of installation tips.

Accessing the Automatic Storage Management Home Page

All Automatic Storage Management (ASM) administration tasks begin with the ASM Home page in Enterprise Manager. The ASM Home page displays:

- The status of the ASM instance
- A chart that shows the used and free space of each disk group
- A list of databases that are serviced by the ASM instance
- Alerts and metrics
- Links to the ASM Performance, Administration, and Configuration pages



Automatic Storage Management: stadh43.us.oracle.com_+ASM

Logged in As SYS

Automatic Storage Management: stadh43.us.oracle.com_+ASM

Data Retrieved February 26, 2005 9:35:29 AM PST

General

Current Status	Up
Up Since	Feb 20, 2005 4:38:49 PM
Availability (%)	100 (Last 24 hours)
Instance Name	+ASM
Version	10.2.0.0.0
Host	stadh43.us.oracle.com
Oracle Home	/u01/app/oracle/oracle/product/10.2.0/db_1
Alert Log	No ORA- errors
Policy Violations	✓ 0

Disk Group Usage (GB)

Serviced Databases

Name	Disk Groups	Space Used (GB)	Availability	Alerts
R2b2	DGROUP1	2.7	100 ⇧	1 0

Alerts

Severity ▽	Category	Name	Message	Alert Triggered
⚠	Alert Log	Alert Log Error Stack	ORA-error stack (07445[08552F41]) logged in /u01/app/oracle/admin/+ASM/bdump/alert_+ASM.log.	Feb 20, 2005 4:41:57 PM

Accessing the ASM Home Page on a Single Instance System

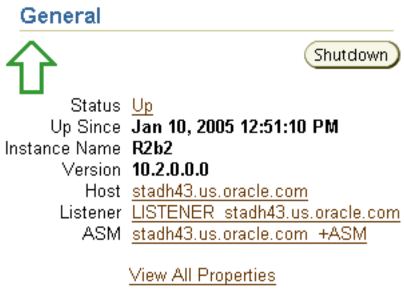
To access the ASM Home page on a single instance system:

1. Log in to Enterprise Manager Database Control as the SYS user, connecting as SYSDBA. Provide the SYS password that is stored in the password file for the ASM instance. This password was set up when the ASM instance was created.

   See "Accessing the Oracle Enterprise Manager Database Control" on page 3-3 for more information.

   > **Note:** You can also log in to Enterprise Manager as a user other than SYS. In this case, Enterprise Manager displays the Automatic Storage Management Login page when you attempt any ASM administration tasks. You must then log in to ASM as the SYS user, connecting as SYSDBA.

2. On the Database Instance Home page, under the General heading, click the link next to the label **ASM**.



General

Status	Up
Up Since	Jan 10, 2005 12:51:10 PM
Instance Name	R2b2
Version	10.2.0.0.0
Host	stadh43.us.oracle.com
Listener	LISTENER_stadh43.us.oracle.com
ASM	stadh43.us.oracle.com_+ASM

View All Properties

Accessing the ASM Home Page On a Real Application Clusters System

To access the ASM Home page on a Real Application Clusters (RAC) system:

1. Log in to Enterprise Manager Database Control on any node that is running the Oracle Management Service (OMS).

   OMS is automatically started on the node where Database Configuration Assistant (DBCA) was run to create the cluster database. Depending on your configuration, OMS may also be running on other nodes.

   See "Accessing the Oracle Enterprise Manager Database Control" on page 3-3 for more information.

2. On the Cluster Database page, under the Instances heading, click the link for the desired ASM instance.

Starting Up and Shutting Down the ASM Instance

To start up or to shut down the ASM instance, all database instances that are serviced by the ASM instance must first be shut down, and you must have operating system user credentials and ASM `SYS` user credentials.

> **Note:** The ASM instance must be started before you start any of the database instances that are serviced by it.

To start or stop the ASM instance:

1. Access the Automatic Storage Management Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-3 for instructions.

2. Under the General heading, click **Startup/Shutdown**
3. Enter the required operating system and ASM user names and passwords, and click **OK**.
4. If you are prompted for the operation to perform, select the desired operation.
5. Click **Yes** on the Startup/Shutdown Confirmation page.

Monitoring Disk Group Space Usage

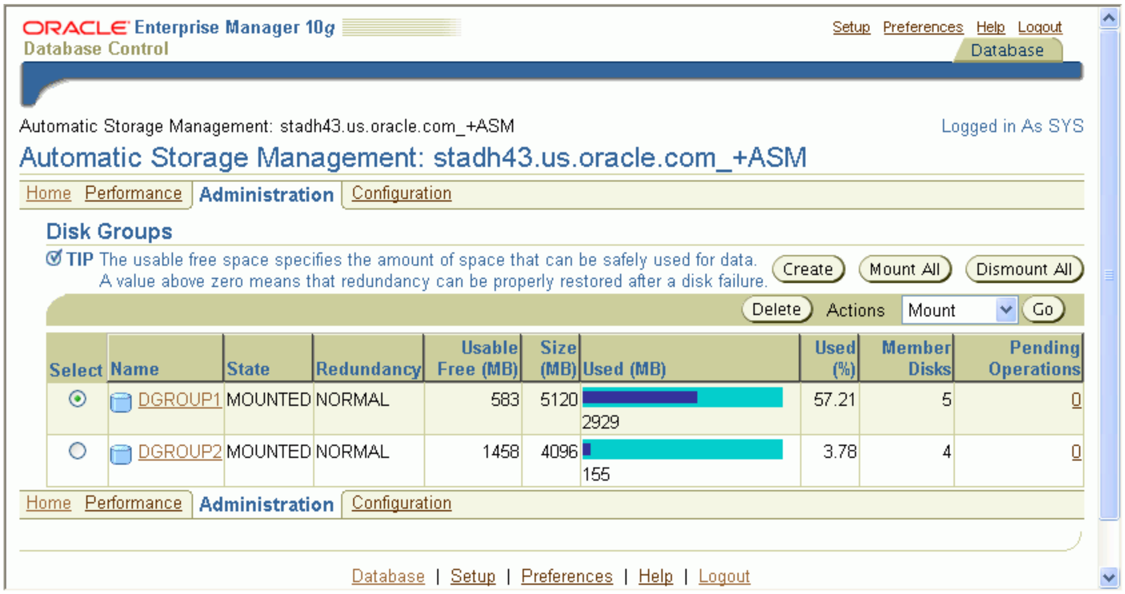
To view Automatic Storage Management (ASM) disk group usage and free space:

1. Access the ASM Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-3 for instructions.

2. Click the **Administration** tab.
3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSDBA`. Provide the `SYS` password that is stored in the password file for the ASM instance. This password was set up when the ASM instance was created.

The ASM Administration page displays all disk groups with their space usage information.


ORACLE Enterprise Manager 10g
Database Control
Setup Preferences Help Logout
Database
Home Performance Administration Configuration
Create
Mount All
Dismount All
Delete
Actions
Mount
Go
Database | Setup | Preferences | Help | Logout

Automatic Storage Management: stadh43.us.oracle.com_+ASM

Logged in As SYS

Automatic Storage Management: stadh43.us.oracle.com_+ASM

Disk Groups

TIP The usable free space specifies the amount of space that can be safely used for data. A value above zero means that redundancy can be properly restored after a disk failure.

Select	Name	State	Redundancy	Usable Free (MB)	Size (MB)	Used (MB)	Used (%)	Member Disks	Pending Operations
●	DGROUP1	MOUNTED	NORMAL	583	5120	2929	57.21	5	0
○	DGROUP2	MOUNTED	NORMAL	1458	4096	155	3.78	4	0

Note: The Usable Free column displays the space in MB that is actually available in the disk group. It takes into account the redundancy level of the disk group, and excludes the space that the disk group reserves for restoring full redundancy for all files in the disk group after a disk failure. See *Oracle Database Administrator's Guide* for more information.

Creating a Disk Group

You may want to create additional Automatic Storage Management (ASM) disk groups to:

- Have disk groups with different redundancy levels (normal, high, or external), depending on availability requirements and storage system capabilities.
- Separate different classes of storage (for example, SCSI drives and SATA drives) into different disk groups. (Disks in a disk group should have similar size and performance characteristics.)
- Store the flash recovery area in a separate disk group from the database.

To create a disk group:

1. Access the ASM Home page.

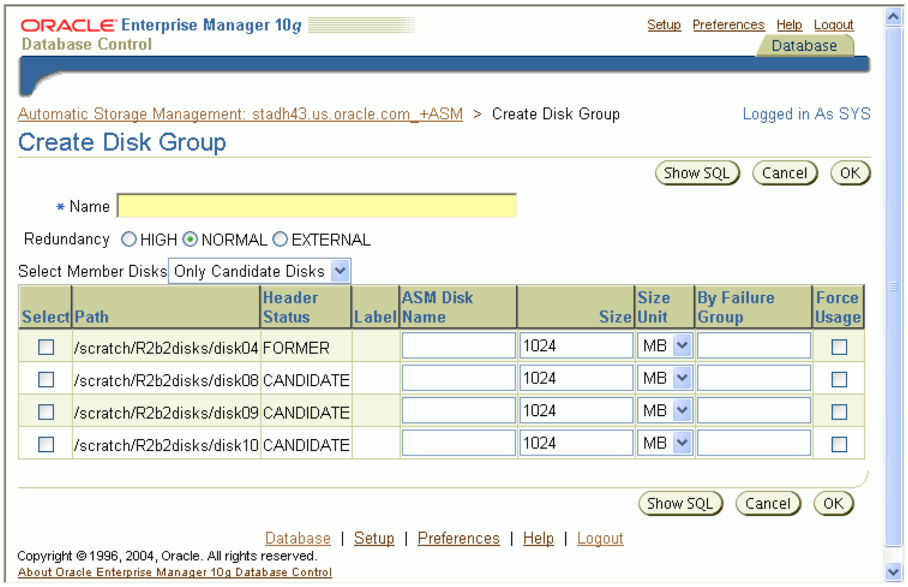
   See "Accessing the Automatic Storage Management Home Page" on page A-3 for instructions.

2. Click the **Administration** tab.

3. If the Automatic Storage Management Login page appears, log in as the SYS user, connecting as SYSDBA. Provide the SYS password that is stored in the password file for the ASM instance. This password was set up when the ASM instance was created.

4. On the ASM Administration page, under the Disk Groups heading, click **Create**.

The Create Disk Group page appears. It displays a list of ASM disks that are available to be added to a disk group. This includes disks with the header status of `CANDIDATE`, `PROVISIONED`, or `FORMER`.



Automatic Storage Management: stadh43.us.oracle.com_+ASM > Create Disk Group

Logged in As SYS

Create Disk Group

* Name

Redundancy ○ HIGH ● NORMAL ○ EXTERNAL

Select Member Disks Only Candidate Disks

Select	Path	Header Status	Label	ASM Disk Name	Size	Size Unit	By Failure Group	Force Usage
□	/scratch/R2b2disks/disk04	FORMER			1024	MB		□
□	/scratch/R2b2disks/disk08	CANDIDATE			1024	MB		□
□	/scratch/R2b2disks/disk09	CANDIDATE			1024	MB		□
□	/scratch/R2b2disks/disk10	CANDIDATE			1024	MB		□

Copyright © 1996, 2004, Oracle. All rights reserved.

About Oracle Enterprise Manager 10g Database Control

5. If you want to display not only disks that are available to be added to a disk group, but all ASM disks, including disks that already belong to a disk group (header status = `MEMBER`), select **All Disks** from the Select Member Disks drop-down list.

   The page is redisplayed with the complete list of ASM disks that were discovered by the ASM instance.

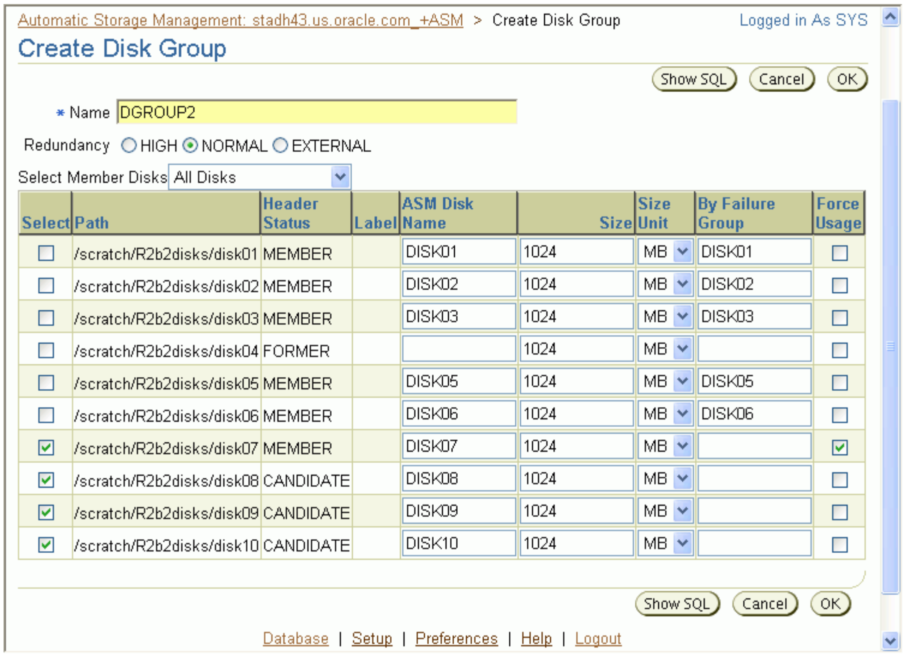
6. Enter the disk group name, and select redundancy type (high, normal, or external).

7. Select disks as follows:

   - Select the check box to the left of each disk that you want to include in the new disk group.
   - If you want to force the inclusion of a disk in the disk group, select the **Force Usage** check box for that disk.

     Caution: The Force Usage check box causes the disk to be added to the new disk group even if the disk already belongs to another disk group and has valid database data. This data will be lost. You must be certain that you are selecting a disk that can legitimately be added to the disk group. See *Oracle Database Administrator's Guide* for a discussion of the `FORCE` option.

   - Optionally enter an ASM name for each selected disk. (ASM provides a name if you do not.)


Automatic Storage Management: stadh43.us.oracle.com_+ASM > Create Disk Group
Logged in As SYS
Show SQL
Cancel
OK

Create Disk Group

* Name DGROUP2

Redundancy ○ HIGH ● NORMAL ○ EXTERNAL

Select Member Disks All Disks

Select	Path	Header Status	Label	ASM Disk Name	Size	Size Unit	By Failure Group	Force Usage
□	/scratch/R2b2disks/disk01	MEMBER		DISK01	1024	MB	DISK01	□
□	/scratch/R2b2disks/disk02	MEMBER		DISK02	1024	MB	DISK02	□
□	/scratch/R2b2disks/disk03	MEMBER		DISK03	1024	MB	DISK03	□
□	/scratch/R2b2disks/disk04	FORMER			1024	MB		□
□	/scratch/R2b2disks/disk05	MEMBER		DISK05	1024	MB	DISK05	□
□	/scratch/R2b2disks/disk06	MEMBER		DISK06	1024	MB	DISK06	□
☑	/scratch/R2b2disks/disk07	MEMBER		DISK07	1024	MB		☑
☑	/scratch/R2b2disks/disk08	CANDIDATE		DISK08	1024	MB		□
☑	/scratch/R2b2disks/disk09	CANDIDATE		DISK09	1024	MB		□
☑	/scratch/R2b2disks/disk10	CANDIDATE		DISK10	1024	MB		□

8. Click **OK** to create the disk group.

   See Also: The discussion of the V$ASM_DISK view in *Oracle Database Reference* for information on the various header statuses of ASM disks.

Dropping a Disk Group

When you drop a disk group, you delete the disk group, and optionally all of its files. You cannot drop a disk group if any of its database files are open. After dropping a disk group, you can add its member disks to other disk groups or use them for other purposes.

One reason to drop a disk group is to change redundancy (normal, high, or external). Because you cannot change the redundancy of a disk group, you must drop the disk group and then re-create it with the proper redundancy. In this case, you must back up or move disk group data before you drop the disk group.

To drop a disk group:

1. Access the ASM Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-3 for instructions.

2. Click the **Administration** tab.

3. If the Automatic Storage Management Login page appears, log in as the SYS user, connecting as SYSDBA. Provide the SYS password that is stored in the password file for the ASM instance. This password was set up when the ASM instance was created.

4. On the ASM Administration page, under the Disk Groups heading, click in the **Select** column to select the disk group that you want to drop.

5. Click **Delete**.

   A confirmation page appears.

6. If you want to delete the contents of the disk group, click **Advanced Options**, ensure that the `INCLUDING CONTENTS` option is selected, and then click **OK**.
7. On the confirmation page, click **Yes**.

Adding Disks to a Disk Group

You add disks to a disk group to increase the total amount of storage space in a disk group. You can add one or multiple disks in a single operation. ASM then rebalances the disk group so that data is evenly distributed on all disks, including the newly added disks.

You can control the **power** of the rebalance operation, which is a number from 0 to 11. The higher the number, the faster the rebalance operation completes. Lower numbers cause rebalancing to take longer, but consume fewer processing and I/O resources. This leaves these resources available for the database. The default value of 1 minimizes disruption to the database. A value of 0 blocks the rebalance operation altogether. Manual or automatic rebalance can then occur at a later time. (You may want to postpone rebalancing because you want to wait for a time when there are fewer demands on the database, or because you want to add more disks or drop disks later and want the rebalancing to be done only once for all disk group changes.)

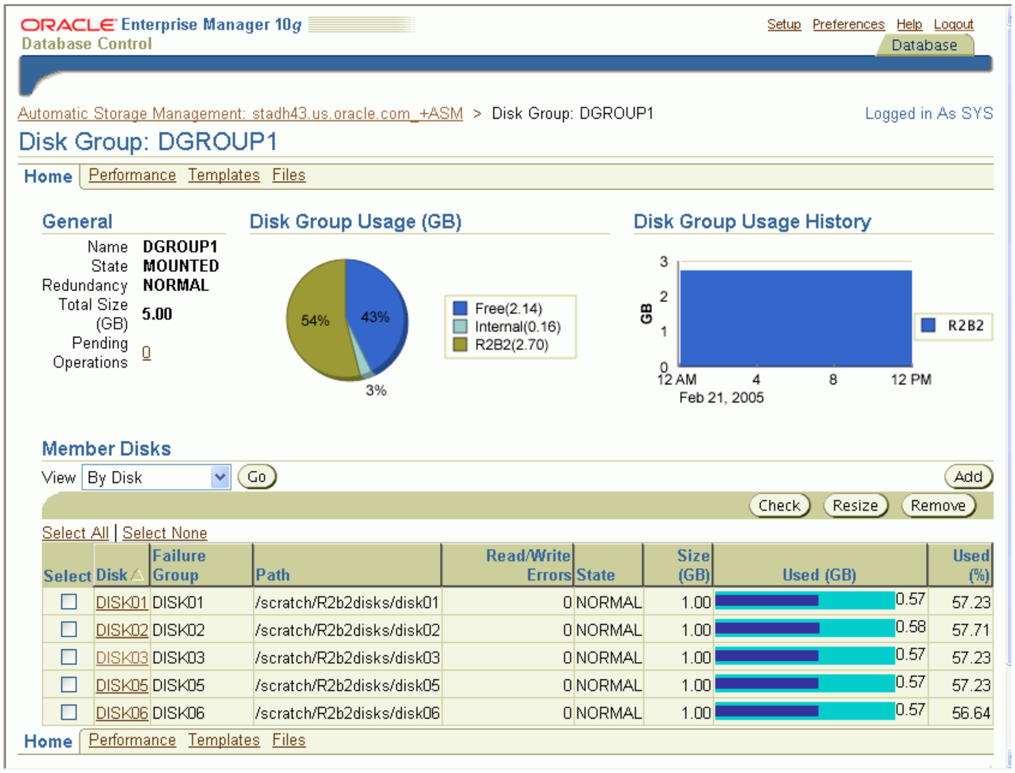
See *Oracle Database Administrator's Guide* for more information about controlling rebalance operations.

To add one or more disks to a disk group:

1. Access the ASM Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-3 for instructions.
2. Click the **Administration** tab.
3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSDBA`. Provide the `SYS` password that is stored in the password file for the ASM instance. This password was set up when the ASM instance was created.
4. On the ASM Administration page, under the Disk Groups heading, click a link in the Name column to select the disk group to which you want to add disks.

   The Disk Group page appears, displaying a list of disks that are already in the disk group.



Automatic Storage Management: stadh43.us.oracle.com_+ASM > Disk Group: DGROUP1

Logged in As SYS

Disk Group: DGROUP1

General

Name	DGROUP1
State	MOUNTED
Redundancy	NORMAL
Total Size (GB)	5.00
Pending Operations	0

Disk Group Usage (GB)

Disk Group Usage History

Member Disks

Select	Disk △	Failure Group	Path	Read/Write Errors	State	Size (GB)	Used (GB)	Used (%)
□	DISK01	DISK01	/scratch/R2b2disks/disk01	0	NORMAL	1.00	0.57	57.23
□	DISK02	DISK02	/scratch/R2b2disks/disk02	0	NORMAL	1.00	0.58	57.71
□	DISK03	DISK03	/scratch/R2b2disks/disk03	0	NORMAL	1.00	0.57	57.23
□	DISK05	DISK05	/scratch/R2b2disks/disk05	0	NORMAL	1.00	0.57	57.23
□	DISK06	DISK06	/scratch/R2b2disks/disk06	0	NORMAL	1.00	0.57	56.64

5. Click **Add**.

   The Add Disks page appears. It displays a list of ASM disks that are available to be added to the disk group. This includes disks with the header status of `CANDIDATE`, `PROVISIONED`, or `FORMER`.

6. If you want to display not only disks that are available to be added to a disk group, but all ASM disks, including disks that already belong to a disk group (header status = `MEMBER`), select **All Disks** from the Select Member Disks drop-down list.

   The page is redisplayed with the complete list of ASM disks that were discovered by the ASM instance.

7. Optionally change the rebalance power by selecting from the Rebalance Power drop-down list.

8. Select disks as follows:

   - Select the check box to the left of each disk that you want to add to the disk group.
   - If you want to force the inclusion of a disk in the disk group, select the **Force Usage** check box at the right.

     Caution: The Force Usage check box causes the disk to be added to the disk group even if the disk already belongs to another disk group and has valid database data. This data will be lost. You must be certain that you are selecting a disk that can legitimately be added to the disk group. See *Oracle Database Administrator's Guide* for a discussion of the `FORCE` option.

  - Optionally enter an ASM name for each disk. (ASM provides a name if you do not.)

9. Click **OK** to add the selected disks.

Dropping Disks from a Disk Group

When you drop a disk from the disk group, the disk group is rebalanced by moving all of the file extents from the dropped disk to other disks in the disk group. ASM then releases the disk, and you can then add it to another disk group or use it for other purposes. You can drop one or multiple disks in a single operation. You can also optionally set rebalance power for the drop operation.

The following is a list of possible reasons for dropping a disk:

- A disk is starting to fail, and you need to replace it.
- You want to upgrade a disk.
- You want to reallocate the disk to a different disk group, or reallocate the disk to a different storage system.

> **Note:** Dropping disks from a disk group only logically deletes the disks from the disk group. It does not delete the disk contents. However, the contents are lost when you add the disk to a new disk group.

To drop one or more disks from a disk group:

1. Access the ASM Home page.

   See "Accessing the Automatic Storage Management Home Page" on page A-3 for instructions.

2. Click the **Administration** tab.
3. If the Automatic Storage Management Login page appears, log in as the `SYS` user, connecting as `SYSDBA`. Provide the `SYS` password that is stored in the password file for the ASM instance. This password was set up when the ASM instance was created.
4. On the ASM Administration page, under the Disk Groups heading, click a link in the **Name** column to select the disk group from which you want to drop disks.

   The Disk Group page appears.

5. Under the Member Disks heading, in the Select column, select the check boxes for the disks that you want to drop, and then click **Remove**.

   A confirmation page appears.

6. If you want to change rebalance power, or if you want set the `FORCE` option for dropping disks, do the following:

   - Click **Advanced Options**.
   - On the Advanced Options page, optionally select the `FORCE` option or select a rebalance power in the drop-down list.
   - Click **OK** to return to the confirmation page.

Note: You may need the FORCE option if ASM cannot read from or write to the disk. See *Oracle Database Administrator's Guide* for more information on the FORCE option.

7. Click **Yes** to drop the disks.

   The Disk Group page returns, and displays a state of DROPPING for each disk being dropped.

8. Refresh the page until the dropped disks no longer appear in the disk group.

Caution: You cannot reuse or disconnect the dropped disks until the drop and rebalance operations are complete—that is, until the dropped disks no longer appear in the disk group. See *Oracle Database Administrator's Guide* for more information, and for other caveats for dropping disks.

Backing Up ASM-Managed Files

Oracle recommends that you use RMAN to back up files managed by ASM. The procedures for using RMAN are documented in Chapter 9, "Performing Backup and Recovery".

Automatic Storage Management: Oracle by Example Series

Oracle by Example (OBE) has a series on the *Oracle Database 2 Day DBA* book. This OBE steps you through the tasks in this appendix and includes annotated screen shots.

To view the Automatic Storage Management OBE, point your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/asm/asm.htm
```

B

Overview of Real Application Clusters

This appendix provides an overview of Oracle Real Application Clusters (RAC). This chapter contains the following topics:

- What is Real Application Clusters?
- Real Application Clusters Database Configuration Scenario
- Differences in Administering Real Application Clusters and Single-Instance Oracle Databases
- Monitoring Oracle Real Application Clusters with Enterprise Manager
- Starting and Stopping Real Application Clusters Databases and Database Instances
- Managing Real Application Clusters Backup and Recovery

What is Real Application Clusters?

Real Application Clusters (RAC) is a technology that links one or more individual computers so that they function as one system. RAC enables each computer, or node, that is a member of a cluster database to share access to the Oracle database. If one cluster node fails or is taken offline, then the other cluster nodes continue operating and the entire RAC database remains available. This means that two or more inexpensive computers appear to applications as if they were a much more powerful, and more expensive, computer.

To increase the performance of a two-node RAC database, you can add cluster nodes. Each additional node can help speed up application processing, support more users or processes, or both. In addition, you can also add cluster nodes to increase the availability and reliability of a two-node RAC database. The more nodes that your RAC environment has, the less the effect that the loss of any individual node has on the database.

Note: You can include up to four single-processor nodes or two dual-processor nodes in an Oracle Database Standard Edition installation with RAC. You must upgrade to the Oracle Database Enterprise Edition to add additional nodes.

Real Application Clusters Database Configuration Scenario

A RAC database requires three components: cluster nodes, shared storage, and Oracle Clusterware. Although you can choose how many nodes your cluster should have and what type of shared storage to use, this appendix is based on one specific configuration, which is the recommended configuration for two-computer, or two-node, clusters. This recommended two-node configuration uses Automatic Storage Management (ASM) for storage and Recovery Manager (RMAN) for the backup and recovery strategy. In addition, the instructions for the Enterprise Edition interface tasks use the Enterprise Manager Database Control console. This appendix does not provide cluster database installation and configuration procedures.

> **See Also:** The Oracle Real Application Clusters installation guide for your operating system and the *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide*

Differences in Administering Real Application Clusters and Single-Instance Oracle Databases

Most administration tasks are the same for Oracle single-instance and RAC databases. This appendix provides additional instructions for some of the RAC-specific database administration tasks, as well as some recommendations for managing RAC databases.

Monitoring Oracle Real Application Clusters with Enterprise Manager

This section describes how Oracle Enterprise Manager Console enables you to monitor your RAC database at the cluster level, for a high-level view of activity, and at the instance level, where tuning is similar to single-instance database tuning. This section contains the following topics:

- Overview of the Cluster Database Home Page
- Viewing Real Application Clusters Database Targets
- Navigating to Oracle Enterprise Manager Cluster-Related Pages
- Viewing and Modifying Real Application Clusters Initialization Parameters

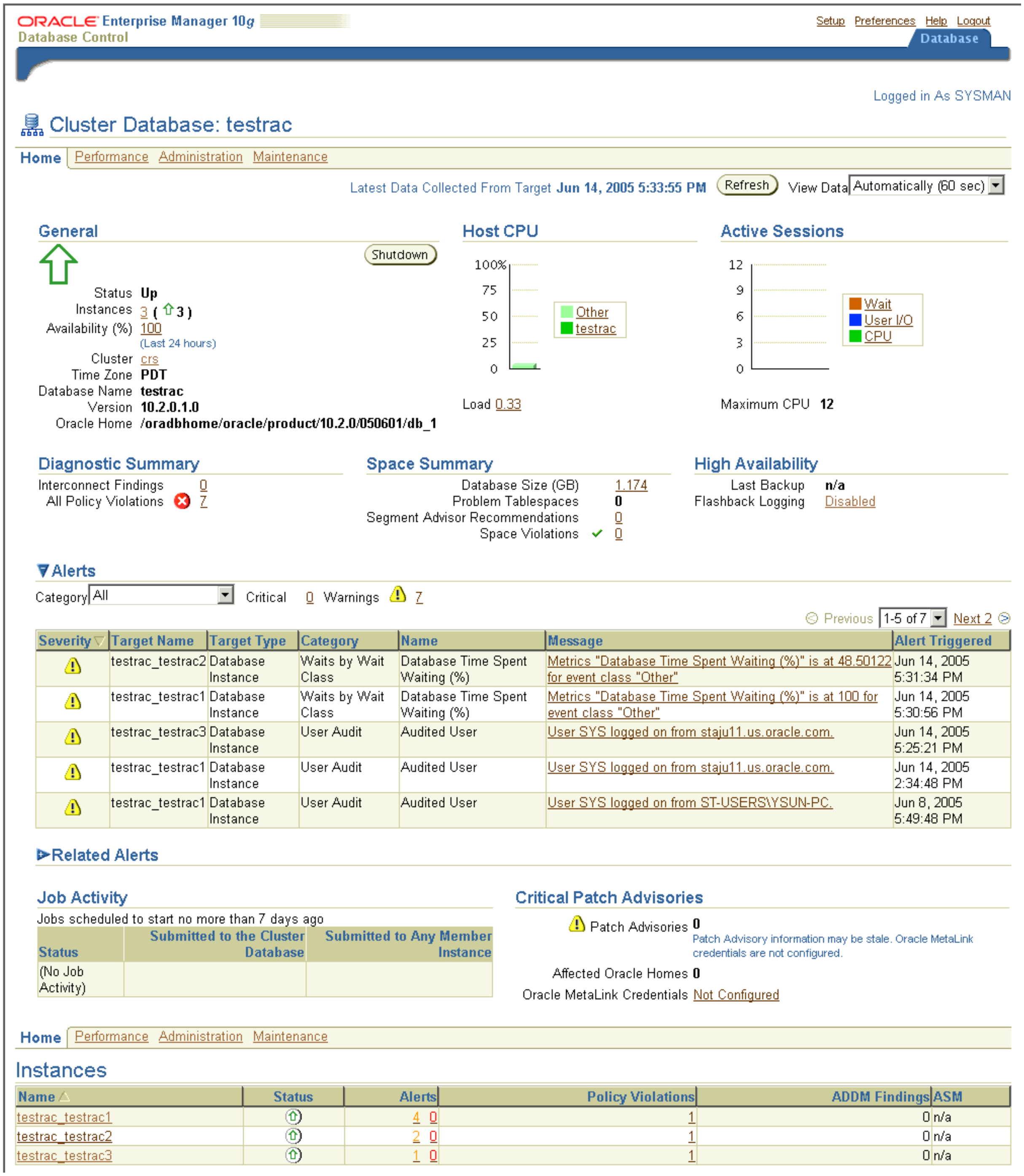
> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more detailed information about RAC database management and administration with Enterprise Manager

Overview of the Cluster Database Home Page

Log in to the RAC database Oracle Enterprise Manager Console from a client browser as described in the section "Starting and Stopping the Oracle Enterprise Manager Console" in Chapter 3, "Getting Started with Oracle Enterprise Manager". When you log in to the Oracle Enterprise Manager Console, Enterprise Manager displays the Cluster Database Home page for the RAC database.

The Cluster Database Home page looks similar to a single-instance Database Control Home Page. However, on the Cluster Database Home page, Oracle Enterprise Manager displays the system state and availability of the entire RAC database. This includes a summary about alerts and job activity as well as links to all of the database and ASM instances.

The Cluster Database Home page shows the status as `UP` for any database instances that are running. The page shows the statuses for non-running instances as `DOWN`. You can start the non-running instances or stop the running instances as needed using the **Startup** and **Shutdown** buttons respectively that appear in the General section of the page.



Logged in As SYSMAN

Cluster Database: testrac

Home | Performance | Administration | Maintenance

Latest Data Collected From Target **Jun 14, 2005 5:33:55 PM** Refresh View Data Automatically (60 sec)

General

Shutdown

Status **Up**
Instances 3 (⇧3)
Availability (%) 100
(Last 24 hours)
Cluster crs
Time Zone **PDT**
Database Name **testrac**
Version **10.2.0.1.0**
Oracle Home **/oradbhome/oracle/product/10.2.0/050601/db_1**

Host CPU

Load 0.33

Active Sessions

Maximum CPU **12**

Diagnostic Summary

Interconnect Findings 0
All Policy Violations ✖ 7

Space Summary

Database Size (GB) 1,174
Problem Tablespaces **0**
Segment Advisor Recommendations 0
Space Violations ✓ 0

High Availability

Last Backup **n/a**
Flashback Logging Disabled

▼Alerts

Category All Critical 0 Warnings ⚠ 7

Previous 1-5 of 7 Next 2

Severity ▽	Target Name	Target Type	Category	Name	Message	Alert Triggered
⚠	testrac_testrac2	Database Instance	Waits by Wait Class	Database Time Spent Waiting (%)	Metrics "Database Time Spent Waiting (%)" is at 48.50122 for event class "Other"	Jun 14, 2005 5:31:34 PM
⚠	testrac_testrac1	Database Instance	Waits by Wait Class	Database Time Spent Waiting (%)	Metrics "Database Time Spent Waiting (%)" is at 100 for event class "Other"	Jun 14, 2005 5:30:56 PM
⚠	testrac_testrac3	Database Instance	User Audit	Audited User	User SYS logged on from staju11.us.oracle.com.	Jun 14, 2005 5:25:21 PM
⚠	testrac_testrac1	Database Instance	User Audit	Audited User	User SYS logged on from staju11.us.oracle.com.	Jun 14, 2005 2:34:48 PM
⚠	testrac_testrac1	Database Instance	User Audit	Audited User	User SYS logged on from ST-USERS\YSUN-PC.	Jun 8, 2005 5:49:48 PM

▶Related Alerts

Job Activity

Jobs scheduled to start no more than 7 days ago

	Submitted to the Cluster Database	Submitted to Any Member Instance
Status		
(No Job Activity)		

Critical Patch Advisories

⚠ Patch Advisories **0**
Patch Advisory information may be stale. Oracle MetaLink credentials are not configured.
Affected Oracle Homes **0**
Oracle MetaLink Credentials Not Configured

Home | Performance | Administration | Maintenance

Instances

Name △	Status	Alerts	Policy Violations	ADDM Findings	ASM
testrac_testrac1	⇧	4 0	1	0	n/a
testrac_testrac2	⇧	2 0	1	0	n/a
testrac_testrac3	⇧	1 0	1	0	n/a

Click the **Performance** tab to see detailed RAC database metrics. If critical or warning alerts appear in the Severity column, then you can analyze them by clicking directly on the alert.

The **Alerts** table is displayed under the metrics section on the Cluster Database Home Page. The **Alerts** table lists cluster-wide database alerts in order of severity, sorted by database components. For more information about an alert, click the text in the **Message** column. This takes you to the specific page that measures the system state of the process that triggered the alert.

On the Cluster Database Home Page level, you can select the **Performance**, **Administration**, and **Maintenance** tabs at the top of the page or you can select the same tabs from just above the **Instances** table. These three links enable you to perform clusterwide administration tasks. You can obtain help to understand the task options and metric definitions by clicking the **Help** link next to the **Database** tab. After you navigate to individual metrics displays, or to return to the Cluster Database Home Page, click the **Database** tab in the upper right-hand corner of the page.

Viewing Real Application Clusters Database Targets

On the Cluster Database Page, a **target** can be a cluster, a database instance, or an Oracle utility, such as a listener. The console obtains metrics that provide information about the current states of the targets. The Cluster Database page displays these metrics in separate section headings for each target.

On the Cluster Database Cluster Home page, Oracle Enterprise Manager displays high-level metrics about the state of the cluster. You can navigate to more detailed pages that display metrics about the target that you have selected by clicking the names of individual targets on the page. Links to the database instances and ASM instances appear at the bottom of the Cluster Home page. You can easily view all of the instance targets by clicking these links without navigating to the database and ASM instance detailed home pages.

Navigating to Oracle Enterprise Manager Cluster-Related Pages

You can monitor various activities in your cluster database environment with the other pages in Oracle Enterprise Manager that this section describes. The Cluster Database Performance page, that you access by clicking the **Performance** tab from the Cluster Database Home page, provides a brief summary of the performance statistics for all of the instances of the database. In other words, statistics are rolled up across all of the instances in a RAC database and shown on the Database Performance page.

Using the links that are next to the charts at the bottom of the page, you can drill down to perform the following tasks:

- Identify and resolve the causes of performance issues
- Decide whether you need to add or re-distribute resources
- Tune your SQL plan and schema for better optimization

Note: You can drilldown to problematic instances by way of the roll up pages. This enables you to directly access instances to perform tuning or diagnostics without navigating through individual instance performance pages.

For performance information about individual instances, you can use the Cluster Database Instance Performance page. To reach this page for a particular instance, click the **Performance** tab on the Cluster Database Home page, then click the name of the instance that you want to monitor at the bottom of the page. You can drill down from links beside the charts or at the bottom of the Database Instance Performance page to perform the following tasks:

- Identify and resolve the cause of performance issues
- Decide whether you need to add or re-distribute resources
- Tune your SQL plan and schema for better optimization

Use the Cluster Performance Page to see utilization statistics, such as CPU, Memory, and Disk reads and writes, during the past hour for all of the nodes of your cluster. Access the Cluster Performance page by clicking the **Cluster** link in the General section of the Cluster Database Performance page, then click the **Performance** tab from the Cluster Home page. With these statistics, you can determine whether resources need to be added, suspended, or re-distributed. From this page, you can view the CPU, Memory, and Disk I/O charts for each node individually by clicking the host name in the legend to the right of the charts. The resulting Host page also shows statistics for CPU load, memory scan page rate, and longest service time.

The Cluster Interconnects page enables you to monitor interconnects and to identify configuration and transfer rate-related issues, including excess traffic. Use the information on the Cluster Interconnects page to determine how the instances and databases affect the interconnect load. Sometimes you can immediately identify interconnect delays that are due to applications that are outside of the Oracle database. To access the Cluster Interconnects page, click the **Cluster** link in the General section of the Cluster Database Home page, then click the **Interconnects** tab. The tables on the Interconnects page enable you to do the following:

- View all interfaces that are configured across the cluster
- View statistics for the interfaces, such as absolute transfer rates and errors
- Determine the type of interfaces, such as private or public
- Determine whether the instance is using a public or private network
- Determine which database instance is currently using which interface
- Determine how much the instance is contributing to the transfer rate on the interface

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for details about the use and interpretation of the Oracle Enterprise Manager pages discussed in this section

Viewing and Modifying Real Application Clusters Initialization Parameters

Managing initialization parameters across an Oracle RAC database is essentially the same as managing them for a single-instance Oracle database. Note the following differences for parameters in RAC databases:

- Parameters that are cluster-specific are identified in the Category column with the designation **Cluster Database**
- Parameters that are duplicated on each instance in the RAC database are identified in the Instance column with an asterisk (*)
- Parameters that you can set differently on each instance of a RAC database are listed by instance number

The administration of initialization parameters in RAC environments is slightly different from single-instance database parameter administration. For example, if you change a parameter setting that is marked by an *asterisk*, which indicates that the parameter is a cluster-wide database initialization parameter, then you change that parameter's setting for all of the instances in your RAC database. If you change a *numbered* initialization parameter, which indicates an instance-specific initialization parameter, then the change only affects the instance that corresponds to the number; the change does not affect the parameter's settings on other database instances.

See Also: *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about initialization parameters in RAC environments

Starting and Stopping Real Application Clusters Databases and Database Instances

To start or stop RAC databases using Enterprise Manager, you must first ensure that the Enterprise Manager console is up. If you are not sure how to access the console, then review the section "Starting and Stopping the Oracle Enterprise Manager Console" in Chapter 3, "Getting Started with Oracle Enterprise Manager".

Typically, you start up and shut down the cluster from the Enterprise Manager Cluster Database Home page. By using this page for cluster database startup and shutdown operations, you ensure that all of the instances that belong to the RAC database are in a consistent state. This enables you to more easily manage a RAC database. You can also start and stop individual instances in a RAC database. However, starting and stopping one instance in a RAC database does not stop or start other instances. To completely stop a RAC database, you must shut down all of its instances.

Use Oracle Enterprise Manager to start and stop the entire RAC database, or to start and stop individual instances, as described in the following sections:

- Starting and Stopping Real Application Clusters Databases
- Starting and Stopping Individual Instances in Real Application Clusters Databases

Starting and Stopping Real Application Clusters Databases

The following procedures describe how to start and stop an entire RAC database, as well as each cluster database instance:

1. Log in to Oracle Enterprise Manager Database Control.
2. On the Cluster Database Home page in the General view, click **Startup** if the database is down, or **Shutdown** if the database is up.
3. On the Startup/Shutdown: Specify Credentials page, enter the cluster database host credentials for the database nodes. The host credentials are the user name and password for a user who is a member of the OSDBA or OSOPER system privileges group.
4. On the Startup/Shutdown: Select Operation page, click **Startup All** to start all of the instances, or click **Shutdown All** to stop all of the instances.
5. On the Startup/Shutdown: Confirmation page, click **Yes**.

Starting and Stopping Individual Instances in Real Application Clusters Databases

To start and stop individual instances, navigate to the RAC Database Startup and Shutdown page and select the instance that you want to start or stop. Then start or stop the instance as needed.

Note: You can also start up and shut down individual instances from each instance’s home page. However, it is easier to perform instance startup and shutdown operations directly from the Database Startup and Shutdown page.

Managing Real Application Clusters Backup and Recovery

Managing backup and recovery in RAC database environments is similar to managing backup and recovery in single-instance Oracle database environments. This section provides an overview of some of the additional concepts and procedures for backup and recovery in RAC environments. This section contains the following topics:

- Redo Log Groups and Redo Threads in Real Application Cluster Databases
- Backing Up Real Application Clusters Databases
- Overview of Restore and Recovery Operations for a Real Application Clusters Database
- Using a Flash Recovery Area with a Real Application Clusters Database

Redo Log Groups and Redo Threads in Real Application Cluster Databases

Redo logs contain a record of changes that have been made to datafiles. In a single-instance Oracle database environment, redo logs generated by database datafile changes are stored in two or more redo log file groups. Each of these groups contains a redo log file and possibly one or more mirrored copies of that file. In a RAC database, each instance requires its own set of redo log groups, which is known as its thread of redo. Mirrored copies of the redo log files provide your system with extra protection against data loss that is due to hardware failures or data corruption.

Each redo log thread must contain at least two redo log groups. Oracle recommends that each of your instances has a redo thread that contains the same number of redo log groups and, as with single-instance Oracle databases, each group should contain the same number of members. For example, in a RAC database with two instances, each instance could have a thread of redo that contains five redo log groups. This is a total of ten redo log groups for the database. Each of these groups could contain two members: a redo log and its mirrored copy. If you create your RAC database with the Database Configuration Assistant (DBCA), then your RAC database automatically implements a configuration that meets the Oracle recommendations.

In a RAC database, each instance writes and archives the redo log groups in its own thread in the same manner that single-instance Oracle databases do. This is described in Chapter 6, "Managing Database Storage Structures". However, in recovery mode, the instance performing the recovery is able to read and process all of the redo threads for the database, regardless of which instance generated the redo threads. This enables a running instance to recover the work completed by the failed instance. This also enables end users to continue working without waiting for the failed instance to be restarted. For example, assume that you have a RAC database with two instances, instance A and instance B. If instance A is down, then instance B can read the redo log files both for instance A and B to ensure a successful recovery.

In a RAC database that uses ASM, all of the redo log files reside on shared storage. In addition, each instance has transparent access to the redo log files of all of the other instances in the cluster.

To access the redo log file groups with Enterprise Manager:

1. Go to the Cluster Database Home Page, and click the **Administration** tab.
2. On the Cluster Database Administration page, under the Storage column, select **Redo Log Groups**.

On the Redo Log Groups page, you can create additional redo log groups and add members to the redo log group. The **Thread** column identifies the instance thread to which a redo log file belongs.

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for additional information about redo log threads in a RAC database environment, and "Online Redo Log Files" on page 6-2

Backing Up Real Application Clusters Databases

When you use ASM to manage database files, Oracle recommends that you use RMAN for backup. You must have both database (SYSDBA) credentials and host operating system (OSDBA) credentials to perform backup and recovery.

In addition, if you log in to Enterprise Manager with SYSDBA privileges, any valid host operating system user who has execute permission for the RMAN command line client can perform backups. However, if you log in as a non-SYSDBA user, then you must provide an operating system user name that is a member of the OSDBA group.

To back up a RAC database:

1. Go to the Cluster Database Home Page, and click the **Maintenance** tab.
2. On the Cluster Database Maintenance page, under the Backup/Recovery column, select **Schedule Backup**.
3. Follow the backup procedures outlined in Chapter 9, "Performing Backup and Recovery".

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for additional information about backing up RAC databases

Overview of Restore and Recovery Operations for a Real Application Clusters Database

This section discusses both instance recovery and media recovery. Instance recovery in RAC is automatic. Instance recovery can occur while a database is running. If a RAC instance fails, then a surviving database instance processes the online redo logs generated by the failed instance to ensure that the database contents are in a consistent state. The more nodes that are in a RAC environment, the shorter the duration of the automatic instance recovery. When recovery completes, the Oracle Clusterware attempts to restart the failed instance automatically.

Media recovery is a manual process that occurs while a database is closed, typically to recover and restore database files after catastrophic disk failures, block corruption, or user errors. Media recovery is always done by one instance in the cluster.

With RMAN, the process of recovering and restoring a RAC database is essentially the same as for a single-instance Oracle databases, except that you access RMAN from the **Maintenance** page at the cluster database level, instead of at the instance level.

> **See Also:** *Oracle Database Backup and Recovery Basics* for additional information about basic recovery and restoration procedures and *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide*

To recover a RAC database:

1. Go to the Cluster Database Home Page and click the **Maintenance** tab.
2. On the Cluster Database Maintenance page, under the Backup/Recovery column, select **Perform Recovery**.

3. Follow the recovery procedures outlined in "Performing Restore and Recovery Operations" in Chapter 9, "Performing Backup and Recovery".

Using a Flash Recovery Area with a Real Application Clusters Database

To use a flash recovery area with a RAC database, the recovery area must be shared by all of the instances. To accomplish this, Oracle recommends that you place the flash recovery area on the shared ASM disks. In addition, you must set the DB_RECOVERY_FILE_DEST and DB_RECOVERY_FILE_DEST_SIZE parameters to the same values on all instances.

To use Flash Recovery, you must first configure the flash recovery area, as described in "Configuring the Flash Recovery Area" in Chapter 9, "Performing Backup and Recovery".

> **See Also:** *Oracle Database Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about using Flash Recovery Areas in RAC

Real Application Clusters: Oracle by Example Series

Oracle by Example (OBE) has a series about the *Oracle Database 2 Day DBA* book. This OBE guides you through the tasks that this appendix describes and the OBE includes annotated screen shots. To view the Real Application Clusters OBE, direct your browser to the following location:

```
http://www.oracle.com/technology/obe/10gr2_2day_dba/rac/rac.htm
```

Index

A

adding
- disks to Automatic Storage Management disk group, A-9

administrative user accounts
- SYS, 5-4
- SYSTEM, 5-4

advisors
- Automatic Database Diagnostics Monitor (ADDM), 10-12
- Buffer Cache Advisor, 10-12
- description, 10-11
- Memory Advisor, 10-12
  - using, 10-15
- performance, 10-11
- PGA Advisor, 10-12
- Segment Advisor, 10-12
  - using, 6-10
- Shared Pool Advisor, 10-12
- SQL Access Advisor, 10-12
  - using, 10-14
- SQL Tuning Advisor, 10-12
  - using, 10-13
- Undo Advisor, 6-17, 10-12

archive log files
- description, 6-4

ARCHIVELOG mode, 2-9

ASM
- *see* Automatic Storage Management

Automatic Database Diagnostics Monitor (ADDM)
- description, 10-12
- modifying behavior, 10-10
- overview, 10-2
- viewing performance analysis, 10-9

automatic shared memory configuration
- enabling, 5-9

Automatic Storage Management, A-1
- accessing the Home page, A-3
- administering, A-2
- disk discovery, A-3
- installing, A-3
- monitoring disk group space usage, A-5
- starting up and shutting down, A-5

Automatic Workload Repository, 10-4

B

background processes
- description, 5-3

backing up
- Automatic Storage Management files, A-12

backup
- backup sets, 9-8
- consistent, 9-2
- crosschecking, 9-25, 9-28
- deleting expired backups, 9-29
- deleting obsolete backups, 9-30
- expired, 9-24
- full backup of datafiles, 9-10
- image copies, 9-8
- inconsistent, 9-2
- incremental backup of datafiles, 9-10
- making unavailable, 9-30
- managing, 9-24
- Oracle-suggested backup strategy, 9-15
- overview, 9-1
- settings
  - configuring, 9-8
- validating, 9-16
- whole database backup, 9-11

backup and recovery area
- specifying, 2-9

backup files, 6-8

blackout time periods
- defining, 3-5

blackouts, 3-5

block size
- configuring, 2-10

Buffer Cache Advisor
- description, 10-12

C

central database management
- configuring, 2-8

character sets
- defining, 2-10

client machine
- configuring network, 4-5

compressed tables, 8-6

connect descriptor

description, 4-2
example, 4-2
connect string
description, 4-2
example, 4-2
connection mode
configuring, 2-11
connection privileges
SYSDBA, 5-4
SYSOPER, 5-4
constraints, 8-5
control file
description, 6-2
multiplexing, 6-2
creating a database
steps, 2-4
using standalone DBCA, 2-6
credentials
setting preferred, 3-6
CTXSYS user, 7-2

D

data
loading, 8-13
data integrity, 8-5
database
startup and shutdown, 5-6
Database Configuration Assistant
configuring database options, 2-12
deleting a database, 2-12
managing database templates, 2-12
using to create a database, 2-6
database links
description, 8-26
database objects
naming, 8-2
database resident program units, 8-21
Database Upgrade Assistant
overview, 2-14
restrictions on versions, 2-16
starting, 2-16
steps for upgrading, 2-17
databases
advisors
using, 10-11
alerts, 10-1
responding to alerts, 10-7
setting metrics, 10-6
setting notification rules, 10-8
viewing metrics, 10-4
backup
description, 9-2
backup and recovery, 9-1
configuring options, 2-12
creating with DBCA, 2-6
deleting using DBCA, 2-12
diagnosing performance problems, 10-9
managing
roadmap, 3-7
managing with Enterprise Manager Database Control, 3-1
metrics thresholds, 10-1
monitoring
alert-based, 10-1
monitoring performance, 10-1
performance
advisors, 10-11
recovery
description, 9-2
steps for installing/creating, 2-4
storage structure
logical structures, 6-1
physical structures, 6-1
tuning, 10-1
upgrading, 2-14
users, 7-1
creating, 7-8
using DBCA templates, 2-12
datafiles
description, 6-4
datatypes
CLOB, 8-4
DATE, 8-4
description, 8-3
NUMBER, 8-4
overview, 8-4
VARCHAR2, 8-4
DBCA
see Database Configuration Assistant
dbconsole process
starting, 3-2
DBSNMP user, 7-2
dedicated server mode, 2-11
dedicated server process, 5-5
deleting a database, 2-12
disk discovery
in Automatic Storage Management, A-3
disk group
overview, A-2
disk group, Automatic Storage Management
adding disks to, A-9
creating, A-6
dropping, A-8
dropping disks from, A-11
monitoring disk space usage, A-5
DMSYS user, 7-2
dropping
disks from Automatic Storage Management disk group, A-11

E

emctl
starting, 3-2
Enterprise Manager
backup, 9-11
backup and recovery, 9-6
Enterprise Manager Console
starting, 3-2

Enterprise Manager Database Control
  description, 3-1
  features, 3-1
  granting administrative privileges to use, 3-4
  online help, 3-2
  starting, 3-3
EXAMPLE tablespace, 6-7
  configuring, 2-9

F

failure group, Automatic Storage Management, A-3
file
  loading, 8-13
flash recovery area, 9-3
  configuring, 9-6
  planning, 9-5
  retention policy, 9-4
  specifying, 2-9
flashback drop
  description, 9-23
Flashback Table
  steps, 9-22
flashback table
  description, 9-21
functions
  compile, 8-25
  create, 8-25
  definition, 8-25
  delete, 8-25
  execute, 8-25
  modify, 8-25

G

global database name, 2-2
global indexes, 8-17
grid control
  configuring, 2-8

H

host credentials, B-6

I

indexes
  creating, 8-18
  description, 8-16
  global, 8-17
  local, 8-17
  partitioned, 8-17
initialization parameter file
  server parameter file, 5-2
initialization parameters
  description, 5-2
  memory parameters, 5-9
  viewing, 5-7
installation
  checking prerequisites, 2-1
  choices, 2-2
  steps, 2-4
  using DBCA for database, 2-6
instance
  database instance, 5-1
  management, 5-1
  memory structure, 5-2
  Oracle instance, 5-1
  shutdown, 5-5
  startup, 5-4
isqlplus
  starting, 3-7

L

listener
  starting, 4-5
listener.ora file, 4-1
loading data, 8-13
LOB columns, 8-6
local database management
  configuring, 2-8
local indexes, 8-17

M

MDDATA user, 7-2
MDSYS user, 7-2
media recovery, 9-3
memory
  auto tuning, 5-9
  configuring, 2-10
  managing, 5-9
Memory Advisor
  description, 10-12
  using, 10-15
mirroring, Automatic Storage Management, A-2

N

Net Manager tool, 4-5
network
  configuring client, 4-5
network configuration
  connection requests, 4-1
  description, 4-1
notification of alerts
  alert notification, 3-5

O

OLAPSYS user, 7-2
Oracle listener
  starting, 4-5
Oracle Net
  description, 4-1, 5-5
  listener configuration, 4-1
  mapping methods, 4-2
    directory naming, 4-3
    easy connect naming, 4-3
    external naming, 4-3
    local naming, 4-3

Oracle Net Configuration Assistant
  description, 4-3
oracle system identifier, 2-2
ORDPLUGINS user, 7-2
ORDSYS user, 7-2
OSDBA, B-6
OUTLN user, 7-2

P

package bodies
  compile, 8-24
  create, 8-24
  definition, 8-24
  delete, 8-24
  modify, 8-24
packages
  compile, 8-23
  definition, 8-22, 8-25
  modify, 8-23
partitioned indexes, 8-17
password files, 6-8
PGA Advisor
  description, 10-12
PGA target
  setting, 10-15
preferred credentials, 3-5
  setting, 3-6
privileges
  connection, 5-4
  object, 7-4
  SYSDBA, 9-6
  system, 7-4
procedures
  compile, 8-25
  create, 8-25
  definition, 8-25
  delete, 8-25
  execute, 8-25
  modify, 8-25
profiles
  administering, 7-7
program global area (PGA)
  components, 5-3
  description, 5-3

R

rebalancing, Automatic Storage Management, A-2
recovery
  complete, 9-3
  configuring, 2-9
  flash recovery area, 9-3
  incomplete, 9-3
  media, 9-3
  overview, 9-1
  whole database recovery, 9-18, 9-19
Recovery Manager (RMAN)
  repository, 9-4
redo log files
  description, 6-2
  multiplexing, 6-2, 6-3
roles
  administering, 7-3
  description, 7-4
  granting, 7-11
  viewing, 7-5
rollback segments
  description, 6-4

S

sample schemas
  configuring, 2-9
schema objects
  accessing, 8-1
  database links, 8-26
  description, 8-1
  indexes, 8-16
    creating, 8-18
  sequences, 8-26
  synonyms, 8-26
  tables, 8-3
    constraints, 8-5
    modifying, 8-11
  views, 8-19
Segment Advisor
  description, 10-12
  using, 6-10
sequences
  description, 8-26
server parameter file, 5-2
server processes
  description, 5-5
services
  on Windows, 5-6
SGA target
  setting, 10-15
Shared Pool Advisor
  description, 10-12
shared server mode, 2-11
shared server process, 5-5
shut down database
  Windows, 5-6
shutdown
  database, 5-6
SI_INFORMTN_SCHEMA user, 7-2
SID, 2-2
software
  steps for installing, 2-4
source types
  description, 8-21
  managing, 8-21
SQL Access Advisor
  description, 10-12
  using, 10-14
SQL statements
  about, 3-6
SQL Tuning Advisor
  description, 10-12

using, 10-13
sqlplus
starting, 3-7
standalone subprograms
compile, 8-25
create, 8-25
definition, 8-25
delete, 8-25
execute, 8-25
modify, 8-25
starting database
Windows, 5-6
startup
database, 5-6
synonyms
description, 8-26
SYS user, 5-4, 7-2
SYSAUX tablespace, 6-7
SYSMAN user, 7-2
system global area (SGA)
components, 5-2
description, 5-2
SYSTEM tablespace, 6-7
SYSTEM user, 5-4, 7-2

T

table partitions, 8-6
tables
compressed, 8-6
creating, 8-8
description, 8-3
estimating size, 8-9
LOB columns, 8-6
modifying, 8-11
partitioned, 8-6
tablespaces
creating, 6-8
data block, 6-5
description, 6-5
EXAMPLE, 6-7
modifying, 6-9
Segment Advisor, 6-9
segments
extents, 6-5
SYSAUX, 6-7
SYSTEM, 6-7
TEMP, 6-7
UNDOTBS1, 6-7
USERS, 6-7
TEMP tablespace, 6-7
templates
creating databases with, 2-8
using with DBCA, 2-12
tnsnames.ora file
configuring, 4-5
triggers
compile, 8-25
create, 8-25
definition, 8-25
modify, 8-25

U

undo
advisor, 6-17
automatic undo management, 6-15
managing, 6-14
with Enterprise Manager, 6-16
Undo Advisor
description, 10-12
undo tablespace, 6-15
extending, 6-19
UNDOTBS1 tablespace, 6-7
upgrading a database, 2-17
steps, 2-17
user accounts
description, 7-1
unlocking, 7-11
users
creating, 7-8
CTXSYS, 7-2
DBSNMP, 7-2
DMSYS, 7-2
MDDATA, 7-2
MDSYS, 7-2
OLAPSYS, 7-2
ORDPLUGINS, 7-2
ORDSYS, 7-2
OUTLN, 7-2
provided by Oracle, 7-1
SI_INFORMTN_SCHEMA, 7-2
SYS, 7-2
SYSMAN, 7-2
SYSTEM, 7-2
USERS tablespace, 6-7

V

views
description, 8-19