

Oracle® Database

2 Day DBA

11g Release 2 (11.2)

E10897-09

February 2012

Oracle Database 2 Day DBA 11g Release 2 (11.2)

E10897-09

Copyright © 2004, 2012, Oracle and/or its affiliates. All rights reserved.

Primary Author: Bert Rich

Contributing Author: Kathy Rich, Janet Stern, Steve Fogel, Colin McGregor, Lance Ashdown, Eric Belden, Craig Foch, Sushil Kumar, Roza Leyderman, Paul Needham, and Douglas Williams

Contributor: Bharat Baddepudi, Prasad Bagal, Timothy Chien, Benoit Dageville, Sudip Datta, Kakali Das, Mark Dilman, Prabhaker Gongloor, Shivani Gupta, Pat Huey, Chaitanya Koratamaddi, Balaji Krishnan, Vasudha Krishnaswamy, Rich Long, Venkat Maddali, Matthew McKerley, Ed Miner, Mughees Minhas, Saurabh Pandey, Kant Patel, Hanlin Chien, Mark Townsend, Xiaofang Wang, Kat Weill, and Mike Zampiceni

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xvii
Audience.....	xvii
Documentation Accessibility	xvii
Related Documentation.....	xvii
Conventions	xix
1 Introduction	
About This Guide	1-1
What This Guide Is Not.....	1-1
How to Use This Guide with Related Material.....	1-1
About Oracle Database	1-2
Common Oracle DBA Tasks	1-3
Tools for Administering the Database	1-3
2 Installing Oracle Database and Creating a Database	
Overview of Installing Oracle Database Software and Creating a Database	2-1
Checking Prerequisites.....	2-2
Deciding on Installation Choices	2-2
Install Options	2-3
Installation Method.....	2-3
Installation Type	2-3
Software Installation Directories	2-4
Database File Location	2-4
Database Identifiers.....	2-5
About Advanced Installation.....	2-5
Installing Oracle Database Software	2-6
Creating and Managing a Database with DBCA	2-10
Starting DBCA	2-11
Creating a Database Using DBCA	2-12
Step 2 - Database Templates.....	2-12
Step 3 - Database Identification	2-12
Step 4 - Management Options.....	2-13
Step 5 - Database Credentials.....	2-13
Step 6 - Database File Locations.....	2-13
Step 7 - Recovery Configuration.....	2-13

Step 8 - Database Content	2-15
Step 9 - Initialization Parameters	2-15
Memory	2-15
Sizing	2-16
Character Sets	2-16
Connection Mode.....	2-18
Step 10 - Database Storage	2-18
Step 11 - Creation Options.....	2-18
Changing the Configuration of a Database Using DBCA	2-18
Deleting a Database Using DBCA	2-19
Managing Templates with DBCA.....	2-19
Advantages of Using Templates.....	2-19
Types of Templates.....	2-20
DBCA Templates Provided by Oracle	2-20
Creating Templates Using DBCA.....	2-21
Deleting Templates.....	2-22
Configuring Automatic Storage Management Using Oracle Universal Installer	2-22
Installation: Oracle By Example Series	2-22

3 Getting Started with Database Administration

Managing Your Database: A Roadmap	3-1
Introduction to Oracle Enterprise Manager Database Control	3-2
Configuring the Operating System Environment Variables	3-4
Starting and Stopping the Database Control Console Process	3-5
Accessing the Database Home Page	3-6
Granting Access to Database Control for Nonadministrative Users	3-8
Creating Database Control Administrative Users	3-8
Setting Database Control Preferences.....	3-10
About Database Control Preferences	3-11
Defining Blackout Periods	3-11
Setting Preferred Credentials	3-13
Administering the Database with SQL-Based Management Tools.....	3-14
About SQL.....	3-15
About SQL*Plus.....	3-15
Starting SQL*Plus and Connecting to the Database	3-16
About SQL Developer	3-17
Getting Started with Oracle Enterprise Manager: Oracle By Example Series	3-17

4 Configuring the Network Environment

Understanding Network Configuration	4-1
Oracle Net Listener Configuration	4-2
Client Connections	4-3
Tools for Network Configuration.....	4-4
Viewing Listener Configuration	4-5
Starting and Stopping the Listener.....	4-6
Connecting to an Oracle Database from a Client Computer	4-7
Networking: Oracle By Example Series.....	4-8

5 Managing the Oracle Instance

Overview of the Oracle Instance and Instance Management	5-1
About Initialization Parameters	5-2
About Background Processes	5-3
About Server and Client Processes	5-4
About Instance Memory Structure	5-5
System Global Area	5-5
Program Global Area	5-6
About Instance Startup and Shutdown	5-7
About Administration Privileges for Startup and Shutdown	5-7
About Instance Startup	5-7
About Instance Shutdown	5-8
Shutting Down and Starting Up the Oracle Instance	5-9
Shutting Down and Starting Up Using Oracle Enterprise Manager Database Control	5-9
Shutting Down and Starting Up Using the Windows Services Program	5-11
Viewing and Modifying Initialization Parameters	5-12
Managing Memory	5-15
About Memory Management	5-15
Enabling Automatic Memory Management	5-18
Modifying Memory Settings – Automatic Memory Management	5-19
Modifying Memory Settings – Automatic Shared Memory Management	5-22
Modifying Memory Settings – Automatic PGA Memory Management	5-25
Instances: Oracle By Example Series	5-26

6 Managing Database Storage Structures

About Database Storage Structures	6-1
About Control Files	6-3
About Online Redo Log Files	6-4
About Archived Redo Log Files	6-5
About Rollback Segments	6-5
About Data Files	6-6
About Tablespaces	6-7
Locally Managed Tablespaces	6-8
Tablespace Types	6-8
Tablespace Status	6-9
Autoextend Tablespace	6-9
Encrypted Tablespaces	6-10
About Other Storage Structures	6-10
Viewing Database Storage Structure Information	6-11
Viewing Control File Information	6-11
Viewing Online Redo Log File Information	6-12
Viewing Archived Redo Log File Information	6-12
Viewing Data File Information	6-13
Viewing Tablespace Information	6-14
Performing Common Database Storage Tasks	6-14
Creating a Tablespace	6-14

Modifying a Tablespace	6-18
Setting a Tablespace to Automatically Extend	6-18
Changing Space Usage Alert Thresholds for a Tablespace	6-19
Taking a Tablespace Offline	6-21
Dropping a Tablespace.....	6-22
Reclaiming Unused Space.....	6-22
About Reclaimable Unused Space.....	6-22
About the Segment Advisor.....	6-23
Segment Advisor Recommendations.....	6-24
Viewing Segment Advisor Recommendations and Reclaiming Space.....	6-24
Running the Segment Advisor Manually.....	6-28
Managing the Online Redo Log	6-28
Multiplexing the Online Redo Log.....	6-28
Switching a Log File.....	6-29
Managing Undo Data	6-30
About Undo Data	6-30
About Managing Undo Data	6-31
Viewing Undo Information	6-32
Computing the Minimum Undo Tablespace Size Using the Undo Advisor	6-34
Changing the Undo Tablespace to a Fixed Size.....	6-37
Storage: Oracle By Example Series	6-37

7 Administering User Accounts and Security

About User Accounts	7-1
About User Privileges and Roles.....	7-3
About Administrative Accounts and Privileges.....	7-4
SYS and SYSTEM Users	7-4
SYSDBA and SYSOPER System Privileges.....	7-5
Administering Roles.....	7-6
Viewing Roles	7-6
Example: Creating a Role.....	7-8
Example: Modifying a Role	7-10
Deleting a Role.....	7-10
Administering Database User Accounts.....	7-11
Viewing User Accounts.....	7-12
Example: Creating a User Account.....	7-13
Creating a New User Account by Duplicating an Existing User Account	7-15
Example: Granting Privileges and Roles to a User Account.....	7-16
Example: Assigning a Tablespace Quota to a User Account	7-19
Example: Modifying a User Account	7-20
Locking and Unlocking User Accounts	7-21
Expiring a User Password.....	7-21
Example: Deleting a User Account.....	7-22
Setting the Database Password Policy	7-23
About Password Policies.....	7-23
Modifying the Default Password Policy.....	7-24
Users: Oracle By Example Series.....	7-25

8 Managing Schema Objects

About Schema Objects	8-1
About Schema Object Management Privileges	8-2
Managing Tables	8-2
About Tables	8-3
About Table Types.....	8-3
About Table Column Attributes	8-4
Data Type.....	8-4
NOT NULL Column Constraint.....	8-5
Default Value.....	8-6
Encryption.....	8-6
About Table-Level Constraints	8-6
Other Table Creation Considerations	8-7
User-Defined Types and Large Objects (LOBs)	8-7
Partitioned Tables and Indexes.....	8-7
Physical Storage Attributes	8-8
Compressed Tables.....	8-8
Viewing Tables	8-8
Viewing Table Data.....	8-10
Example: Creating a Table	8-11
Modifying Table Attributes	8-14
Example: Adding Table Columns	8-15
Example: Deleting a Table Column.....	8-15
Example: Adding a New Table Constraint	8-16
Example: Modifying an Existing Table Constraint.....	8-18
Example: Deleting a Table Constraint	8-18
Example: Loading Data into a Table	8-19
Deleting a Table.....	8-21
Managing Indexes	8-21
About Indexes.....	8-21
Indexes and Performance	8-22
Index Attributes	8-22
Viewing Indexes.....	8-24
Example: Creating an Index	8-24
Example: Deleting an Index.....	8-26
Managing Views	8-27
About Views	8-27
Displaying Views	8-28
Example: Creating a View	8-29
Example: Deleting a View.....	8-30
Managing Program Code Stored in the Database	8-30
About Program Code Stored in the Database.....	8-30
Validating (Compiling) Invalid Schema Objects.....	8-31
Working with Other Schema Objects	8-33
Schemas: Oracle By Example Series	8-35

9 Performing Backup and Recovery

Overview of Database Backup and Recovery	9-1
Database Backup and Recovery Concepts	9-3
Consistent and Inconsistent Backups.....	9-3
Media Recovery.....	9-3
Fast Recovery Area.....	9-4
RMAN Repository.....	9-4
Configuring Your Database for Basic Backup and Recovery	9-5
Specifying Credentials for Backup and Recovery Using Database Control.....	9-5
Planning Space Usage and Location for the Fast Recovery Area.....	9-6
About the Backup Retention Policy and the Fast Recovery Area.....	9-6
About the Fast Recovery Area Size.....	9-7
Configuring Recovery Settings.....	9-7
Monitoring Fast Recovery Area Usage.....	9-9
Configuring Backup Settings.....	9-9
About RMAN Backups.....	9-9
Configuring Backup Device Settings.....	9-10
Configuring Backup Policy Settings.....	9-11
Backing Up Your Database	9-12
RMAN Backup Concepts.....	9-12
Full Backups of Data Files.....	9-12
Incremental Backups of Data Files.....	9-13
Incrementally Updated Backups: Rolling Forward Image Copies of Data Files.....	9-13
Backup Tags.....	9-13
Performing and Scheduling Backups Using Database Control.....	9-14
Performing a Whole Database Backup.....	9-14
Using the Oracle Suggested Backup Strategy.....	9-17
About the Oracle Suggested Backup Strategy.....	9-17
Backing Up Your Database Using the Oracle Suggested Disk Backup Strategy.....	9-18
Scheduling Miscellaneous Backup Tasks.....	9-19
Validating Backups and Testing Your Backup Strategy.....	9-19
Validating Selected Backups.....	9-20
Validating Backups for Restore Operations.....	9-20
Displaying Backup Reports	9-21
Managing Backups	9-23
About Backup Management.....	9-23
Cross-Checking Backups.....	9-25
Deleting Expired Backups.....	9-26
Marking Backups as Available or Unavailable.....	9-26
Deleting Obsolete Backups.....	9-27
Performing Oracle Advised Recovery	9-28
About Data Recovery Advisor.....	9-28
Using Data Recovery Advisor.....	9-29
Performing User-Directed Recovery	9-31
Rewinding a Table Using Oracle Flashback Table.....	9-32
Enabling Row Movement on a Table.....	9-32
Performing a Flashback Table Operation.....	9-33

Recovering a Dropped Table Using Oracle Flashback Drop	9-34
Dropping a Table	9-35
Retrieving a Dropped Table	9-36
Rewinding a Database Using Oracle Flashback Database	9-37
Restoring and Recovering the Database	9-38
Backup and Recovery: Oracle By Example Series	9-39

10 Monitoring and Tuning the Database

Proactive Database Monitoring	10-1
About Alerts.....	10-1
Performance Self-Diagnostics: Automatic Database Diagnostic Monitor	10-2
Monitoring General Database State and Workload	10-3
Managing Alerts.....	10-8
Viewing Metrics and Thresholds.....	10-8
Setting Metric Thresholds	10-11
About Responding to Alerts.....	10-13
Clearing Alerts	10-13
Setting Up Direct Alert Notification	10-14
Diagnosing Performance Problems Using ADDM	10-15
Viewing a Summary of ADDM Performance Findings.....	10-15
Responding to ADDM Performance Findings.....	10-18
Running ADDM Manually	10-20
Modifying AWR Snapshot Frequency and Retention	10-22
Using Advisors to Optimize Database Performance	10-23
About Advisors	10-23
About the SQL Advisors	10-25
About the Automatic SQL Tuning Advisor	10-26
Configuring the Automatic SQL Tuning Advisor.....	10-27
Viewing Automatic SQL Tuning Results	10-30
Running the SQL Tuning Advisor.....	10-33
Running the SQL Access Advisor.....	10-35
Optimizing Memory Usage with the Memory Advisors	10-38
About the Memory Advisors	10-38
Example: Obtaining Memory Sizing Advice for ASMM	10-38
Monitoring and Tuning: Oracle By Example Series	10-40

11 Investigating, Reporting, and Resolving Problems

About Database Problems and Incidents	11-1
Investigating, Reporting, and Resolving a Problem	11-2
Roadmap: Investigating, Reporting, and Resolving a Problem	11-3
Task 1 – View Critical Error Alerts in Enterprise Manager	11-4
Task 2 – View Problem Details.....	11-7
Task 3 – (Optional) Create a Service Request.....	11-8
Task 4 – Package and Upload Diagnostic Data to Oracle Support Services	11-10
Task 5 – Track the Service Request and Implement Any Repairs.....	11-12
Task 6 – Close Incidents	11-14

Viewing Problems Using the Enterprise Manager Support Workbench	11-14
---	-------

12 Managing Oracle Database Software

About Software Management and Patch Releases	12-1
Upgrading a Database	12-2
Overview of Database Upgrade Assistant	12-2
Database Versions Supported by DBUA	12-3
Starting DBUA	12-4
Upgrading a Database Using DBUA.....	12-5
Removing Oracle Database Software.....	12-10
Managing Oracle Software: Oracle By Example Series	12-10

A Administering Oracle Automatic Storage Management

About Oracle Automatic Storage Management	A-1
Overview of Disks, Disk Groups, and Failure Groups	A-3
Installing Oracle Automatic Storage Management	A-4
Accessing the Oracle Automatic Storage Management Home Page	A-4
Accessing the Oracle ASM Home Page on a Single-Instance System	A-6
Accessing the Oracle ASM Home Page on an Oracle RAC System.....	A-6
Monitoring Disk Group Space Usage	A-7
Creating a Disk Group	A-8
Dropping a Disk Group	A-10
Adding Disks to a Disk Group	A-10
Dropping Disks from a Disk Group	A-13
Backing Up Oracle ASM-Managed Files.....	A-14
Oracle Automatic Storage Management: Oracle By Example Series	A-14

Index

List of Figures

6-1	Oracle Database Storage Structures	6-2
6-2	Online Redo Log Groups and Their Members	6-4
11-1	Workflow for Investigating, Reporting, and Resolving a Problem	11-3

List of Tables

2-1	DBCA Template Types.....	2-20
2-2	Oracle-Provided DBCA Templates and Their Corresponding Workloads.....	2-20
5-1	Oracle Database Background Processes	5-3
5-2	SGA Components	5-5
5-3	Oracle Database Memory Management Modes	5-17
6-1	Tablespaces and Descriptions	6-7
7-1	Oracle Database Predefined Roles.....	7-3
7-2	System Privileges Granted to the APPDEV Role	7-8
8-1	Common Data Types.....	8-5
10-1	Performance Advisors.....	10-24
10-2	Other Advisors	10-24
11-1	Oracle Advisors that Help Repair Critical Errors	11-13

Preface

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 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.

Audience

Oracle Database 2 Day DBA is for anyone who wants to perform basic administrative tasks with Oracle Database. Only minimal basic knowledge of or experience with database management is required, and a basic knowledge of computers.

In particular, this guide is for the following groups of Oracle users:

- Developers who want to acquire basic database administrator (DBA) skills
- Anyone managing database servers
- Database administrators managing an Oracle database for a small or medium-sized business

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

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

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 Language Reference*
- *Oracle Database Reference*
- *Oracle Database Backup and Recovery User's Guide*
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX or other operating system*
- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Database Installation Guide*

There are also several other 2 Day + books available for a variety of topics:

- *Oracle Database 2 Day + Security Guide*
- *Oracle Database 2 Day + Data Warehousing Guide*
- *Oracle Database 2 Day + Data Replication and Integration Guide*
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database 2 Day + Real Application Clusters Guide*
- *Oracle Database 2 Day + High Availability Guide*
- *Oracle Database 2 Day Developer's Guide*
- *Oracle Database 2 Day + .NET Developer's Guide for Microsoft Windows*
- *Oracle Database 2 Day + PHP Developer's Guide*
- *Oracle Database 2 Day + Application Express Developer's Guide*
- *Oracle Database 2 Day + Java Developer's Guide*

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

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

<http://www.oracle.com/technetwork/community/join/overview/index.html>

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

<http://www.oracle.com/technology/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.
<i>italic</i>	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.

Introduction

As a database administrator (DBA), you are responsible for the overall operation of Oracle Database. This introductory chapter is intended to help orient you to many common DBA tasks, to the tools available to you, and to this guide. The chapter contains the following sections:

- [About This Guide](#)
- [About Oracle Database](#)
- [Common Oracle DBA Tasks](#)
- [Tools for Administering the Database](#)

About This Guide

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 guide is to help you understand the concepts behind Oracle Database, and to help you learn how to perform all common administrative tasks needed to keep the database operational. These tasks include configuring the database, managing memory and storage, managing users, managing database objects such as tables, performing basic troubleshooting, creating backups for your database, performance monitoring activities, and more.

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

What This Guide Is Not

Oracle Database 2 Day DBA is task-oriented. The objective is to describe why and when administrative tasks must be performed. Where appropriate, it describes the concepts necessary to understand and complete a task, assuming the reader has no prior knowledge of the database. This guide is not an exhaustive discussion of all Oracle Database concepts. For this type of information, see *Oracle Database Concepts*.

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

How to Use This Guide with Related Material

This guide is part of a comprehensive set of learning material for administering 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 screenshots. In some cases, the OBE provides additional information to help you complete the task.

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/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/index.html

In addition, Oracle University provides a related instructor-led course called Oracle Database 11g: 2 Day DBA. Contact Oracle University for more information.

About Oracle Database

Oracle Database is a relational database with object and Extensible Markup Language (XML) capabilities. In a relational database, all data is stored in two-dimensional tables that are composed of rows and columns. Oracle Database enables you to store data, update it, and efficiently retrieve it, with a high degree of performance, reliability, and scalability.

Oracle Database is composed of the following elements:

- The Oracle software that you install on your host computer
- The **database**, which is a collection of physical files on one or more disks

The database contains user data, metadata, and control structures. **Metadata**, or data about the data, is the collection of information on disk that permits Oracle software to manage user data. An example of metadata is the data dictionary. Control structures (such as the control file and online redo log files) ensure the integrity, availability, and recoverability of user data.
- The **Oracle instance**, which is composed of the following:
 - The **background processes**, which are the operating system processes or threads that perform the work of accessing, storing, monitoring, and recovering user data, metadata, and control files associated with the database
 - The shared memory areas used by the background processes
- **Server processes** that perform work on behalf of connected users and applications, and the memory and temporary storage used by these processes

Server processes parse and execute SQL statements, and retrieve and return results to the user or application.
- Oracle Net, which is a software layer that enables client applications and Oracle Database to communicate over a network, and the Oracle Net **listener**, which is a process that listens for connection requests from the network.

See Also:

- [Chapter 4, "Configuring the Network Environment"](#)
- [Chapter 5, "Managing the Oracle Instance"](#)
- [Chapter 6, "Managing Database Storage Structures"](#)
- *Oracle Database Concepts* for more information about background processes
- *Oracle Database Reference* for more information about background processes

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 and shutting down the database instance
- Managing the storage structures of the database
- Managing users and security
- Managing database objects, such as tables, indexes, and views
- Backing up the database and performing recovery operations when necessary
- Monitoring the state of the database and taking preventive or corrective action as required
- Monitoring and tuning database performance
- Diagnosing and reporting critical errors to Oracle Support Services

In a small to medium-sized database environment, you might be the sole person performing these tasks. In large, enterprise environments, the job is often divided among several DBAs, each of whom has a specialty, such as database security or database tuning.

See Also:

- *Oracle Database Concepts* for more information about the duties of database administrators

Tools for Administering the Database

The goal of this guide is to enable you to quickly and efficiently create an Oracle database, and to provide guidance in basic database administration.

The following are some products, tools, and utilities you can use to achieve your goals as a database administrator:

- **Oracle Universal Installer**

Oracle Universal Installer (OUI) is a utility that installs your Oracle software and options. It can automatically start Oracle Database Configuration Assistant to install a database.

- **Oracle Database Configuration Assistant**

Oracle Database Configuration Assistant (DBCA) is a utility that 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 new database.

- **Database Upgrade Assistant**

The Database Upgrade Assistant is a tool that guides you through the upgrade of your existing database to a new Oracle Database release.

- **Net Configuration Assistant**

Net Configuration Assistant (NETCA) is a utility that enables you to configure listeners and naming methods, which are critical components of the Oracle Database network.

- **Oracle Enterprise Manager Database Control**

The primary product for managing your database is Oracle Enterprise Manager Database Control (Database Control), a Web-based interface. After you have installed the Oracle Database software, created or upgraded a database, and configured the network, you can use Database Control to manage your database. Database Control also provides an interface for performance advisors and for Oracle utilities such as SQL*Loader and Recovery Manager (RMAN).

Oracle also offers separately licensed Oracle Enterprise Manager management packs, management plug-ins, and other products you can purchase to enhance the capabilities of Oracle Enterprise Manager in specific environments.

See Also:

- *Oracle Database Licensing Information*

Installing Oracle Database and Creating a Database

This chapter describes how to install Oracle Database software and create a database.

If you are using an earlier release of Oracle Database and want to install a later release of the Oracle Database software, then you can *upgrade* your existing Oracle Database and use it with the new release of the database software. See "[Upgrading a Database](#)" on page 12-2.

This chapter contains the following sections:

- [Overview of Installing Oracle Database Software and Creating a Database](#)
- [Installing Oracle Database Software](#)
- [Creating and Managing a Database with DBCA](#)
- [Installation: Oracle By Example Series](#)

Overview of Installing Oracle Database Software and Creating a Database

To install your Oracle Database software, use Oracle Universal Installer (OUI). OUI is a graphical user interface utility that enables you to install new Oracle Database software. Online Help is available to guide you through the installation process.

During the installation process, you are given the opportunity to create a database. If you choose to do so, then OUI automatically starts Oracle Database Configuration Assistant (DBCA) to guide you through the process of creating and configuring a database.

Before you start the installation process, see the following sections for information about prerequisites and installation choices:

- [Checking Prerequisites](#)
- [Deciding on Installation Choices](#)

If you do not create a database during installation, then you must run DBCA at some point after installation to create a database.

Note: After you create a database, either during installation or as a standalone operation, you do not have to create another. Each Oracle instance works with a single database only. Rather than requiring that you create multiple databases to accommodate different applications, Oracle Database uses a single database, and accommodates multiple applications by enabling you to separate data into different *schemas* within the single database. See "[About User Accounts](#)" on page 7-1 for more information about schemas.

Checking Prerequisites

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

- There is a minimum of 1 GB of physical memory.
- Sufficient paging space is available.
- The appropriate service packs or patches for your operating system are installed.
- An appropriate file system format is being used.

See Also:

- *Oracle Database Installation Guide* for your platform for more information about exact requirements

Deciding on Installation Choices

Oracle Universal Installer guides you through an interview phase where you specify your choices for installation and database creation. The exact sequence of steps depends on your operating system. As you progress through the installation, you are presented with choices on how to configure the database.

- [Install Options](#)
- [Installation Method](#)
- [Installation Type](#)
- [Software Installation Directories](#)
- [Database File Location](#)
- [Database Identifiers](#)
- [About Advanced Installation](#)

Install Options

You can choose to create and configure a database, or to only install the database software.

You can create a preconfigured database or a custom-configured database during installation. If you choose not to create a database during installation, then you must run DBCA after installation to create a database.

Preconfigured databases are based on templates that Oracle provides or that you create. Each Oracle-provided template is optimized for a particular workload type. See [Table 2–2, "Oracle-Provided DBCA Templates and Their Corresponding Workloads"](#) on page 2-20 for information about the types of preconfigured databases.

If you choose to use the Desktop Class installation method, then the general purpose database template is used. To create a custom database in which you configure your own database structure, see ["About Advanced Installation"](#).

Note: If you must create a new database, then Oracle recommends that you install a preconfigured database, which is faster and easier. You can customize the database after it has been created.

Installation Method

The installation methods are divided into Desktop Class and Server Class:

- **Desktop Class**—This installation class is most appropriate for laptop or desktop computers. It includes a starter database and requires minimal configuration.
- **Server Class**—This installation class is for servers, such as you would find in a data center, or used to support enterprise-level applications. Choose this installation class if you need access to advanced configuration options.

During a Desktop Class installation, you make only basic choices. For a Server Class installation, you choose either typical installation (where you make only basic choices) or advanced installation.

During a Desktop Class or a typical installation, Oracle Database automatically installs the sample schemas.

Installation Type

When you install Oracle Database during basic and advanced installations, you need answers for the questions listed in this section. OUI provides default values for every choice.

- What type of database edition installation do you want to perform?

Your choices are:

- **Enterprise Edition**—This installation type is the full-featured Oracle Database product that provides data management for enterprise-level applications. It is intended 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 provides core relational database management services and options and includes an integrated set of management tools, replication, Web features, and facilities for building business-critical applications.
- **Standard One Edition**—This installation type is suitable for workgroup, department, or web applications. It provides core relational database management services for single-server environments or highly distributed branch environments. Oracle Standard Edition One includes all the facilities necessary to build business-critical applications.
- **Personal Edition (Microsoft 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.

- What are your database configuration options?

Software Installation Directories

You must specify the directory in which the Oracle Database software is installed, or the location where the product binary files are copied from the installation media. You must choose a location that has enough disk space to contain the software and is accessible by the operating system user performing the installation.

You also specify the location of the Oracle base directory, which is used by all Oracle software products installed on the server. The first time you install Oracle Database software on a server, you are prompted to specify the location of the inventory directory, called `oraInventory`. This directory provides a centralized inventory of all Oracle software products installed on the server. You should use the same value for the Oracle inventory directory each time you perform an Oracle software installation on the server.

Database File Location

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 the file system of your operating 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 about which option to use, then select File System (the default).

- **Automatic Storage Management**—This option enables you to place your data files in Oracle Automatic Storage Management (Oracle ASM) disk groups. If you choose Oracle ASM, then Oracle Database automatically manages database file placement and naming. For environments with a large number of disks, this option simplifies database administration and maximizes performance. Oracle ASM performs software striping and mirroring at the file level for maximum storage flexibility, performance, and availability.

Oracle ASM uses an Oracle ASM instance, which is distinct from the database instance, to configure and manage disk groups. A single Oracle ASM instance can provide storage for multiple databases on the same server.

For more information, see "[About Oracle Automatic Storage Management](#)" on page A-1 and *Oracle Automatic Storage Management Administrator's Guide*.

Note: In past releases, Oracle ASM was installed as part of the Oracle Database installation. With Oracle Database 11g release 2 (11.2), Oracle ASM is part of an Oracle grid infrastructure installation.

To use Oracle ASM for storing database files, you must have installed Oracle ASM and created one or more disk groups before performing the Oracle Database installation.

Database Identifiers

These options include your global database name and system identifier (SID). The **SID** is a unique identifier that is used to distinguish this instance from other Oracle Database instances that you may create later and run concurrently on your system.

The global database name is the full name of the database that uniquely distinguishes it from any other database. The global database name is in the form `database_name.database_domain`, for example `sales.example.com`. The database name portion `sales` is a simple name you call your database. The database domain portion `example.com` specifies the database domain in which the database is located. Together, the database name and domain form the global database name.

About Advanced Installation

During advanced installations using the Server Class method you are prompted to make the additional choices listed in this section, and the choices for a typical installation. The installation process provides default values for every choice.

This guide describes, but does not document, these additional advanced installation choices. For more information, see *Oracle Database Installation Guide* for your platform.

- Product Languages

You choose which language the software should use after it is installed. You can select multiple languages. The default value is English. If you choose a value other than English, it does not change the language used by the installation.

- Database Configuration Type

You select a template to use when configuring the database. You can choose either either General Purpose/Transaction Processing or Data Warehousing.

- Database Configuration Options

You can choose how to configure the database created by the installer. You can select the memory size and management options, the character sets used to store data, the security options for database access, and whether the sample schemas should be installed.

To complete the exercises in this guide 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. Sample schemas are installed by default during typical or Desktop class installations.

- Database Management Options

You specify whether to manage your database centrally or locally using Oracle Enterprise Manager. Central management enables you to manage multiple targets, such as databases and application servers, using a single interface. Using local management you can manage only a single database instance at a time.

To use central management, you must have an Oracle Enterprise Management agent on each *host*, or computer that has Oracle Database software installed on it. These agents are responsible for monitoring all components on that host. If an agent is not found on the local host, then this option is disabled during installation.

If you are setting up a single database for the first time, then it is recommended that you configure local management with Oracle Enterprise Manager, which is

the default. You can later install additional databases and configure central management using Enterprise Manager.

- Recovery Options

You specify whether automated backups should be configured for the database. If you choose this option, you must specify whether the recovery area should be stored on the local file system or in an Oracle ASM disk group. You must also specify the operating system credentials the backup job uses when performing backups.

Note: To use Oracle ASM for recovery area storage, you must have installed Oracle ASM as part of an Oracle grid infrastructure installation and created one or more disk groups before performing the Oracle Database installation.

- Schema Passwords

When you create a database, certain administrative user accounts are created automatically. You are prompted to enter the passwords for the `SYS`, `SYSTEM`, `SYSMAN`, and `DBSNMP` accounts, which enable you to manage and administer the database. You can use the same password for each account, or specify passwords for each account individually. If you do not enter a secure password, you will receive a warning message during installation.

- Operating System Groups

Administrative access to the database is granted by membership in certain operating system groups. You can choose the operating system group to be used for `SYSDBA` access (typically `dba`) and `SYSOPER` access (typically `oper`).

The `SYSDBA` group identifies operating system user accounts that have database administrative privileges and can log in with `SYSDBA` access. The `SYSOPER` group is an optional group for users that should have limited database administrative privileges. See "[SYSDBA and SYSOPER System Privileges](#)" on page 7-5 for more information about these groups and privileges.

Installing Oracle Database Software

This section briefly describes the steps for a desktop-class installation. Most steps are common to all platforms and involve running the Oracle Universal Installer. Platform-specific steps are noted. For further assistance, consult the online Help or the *Oracle Database Installation Guide* for your platform.

Note: The following steps describe the Oracle Universal Installer workflow for a host computer that has no previous Oracle software installed. If your host computer has Oracle software installed, then you may see a different workflow.

To perform a basic installation:

1. Log on to your computer as a member of the administrative group that is authorized to install Oracle Database 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 and permissions to install new software.

2. Do one of the following:

- If you are installing from distribution media, then insert the distribution media for the database into your computer.

The Autorun feature opens the Select a Product to Install window automatically.

- If you downloaded the installation software from the Oracle Web site, then follow the instructions on the site to run the Oracle Universal Installer. Or, see the *Oracle Database Installation Guide* for your platform.

3. The first window that appears is the Configure Security Updates window. To receive notifications about security issues via e-mail, enter your e-mail address in the Email text field. To receive security updates from My Oracle Support, enter the e-mail address registered with My Oracle Support, select the **I wish to receive security updates...** option, and enter your My Oracle Support password.

Click **Next** to continue.

The Select Installation Option window appears.

4. Choose the **Create and configure a database** option. Or, you also have the option of choosing to only install the database software, but then you must create a database in an additional step after the software is installed. If you are currently using a previous version of Oracle Database, choose Upgrade an existing database. After you have chosen an option, click **Next**.

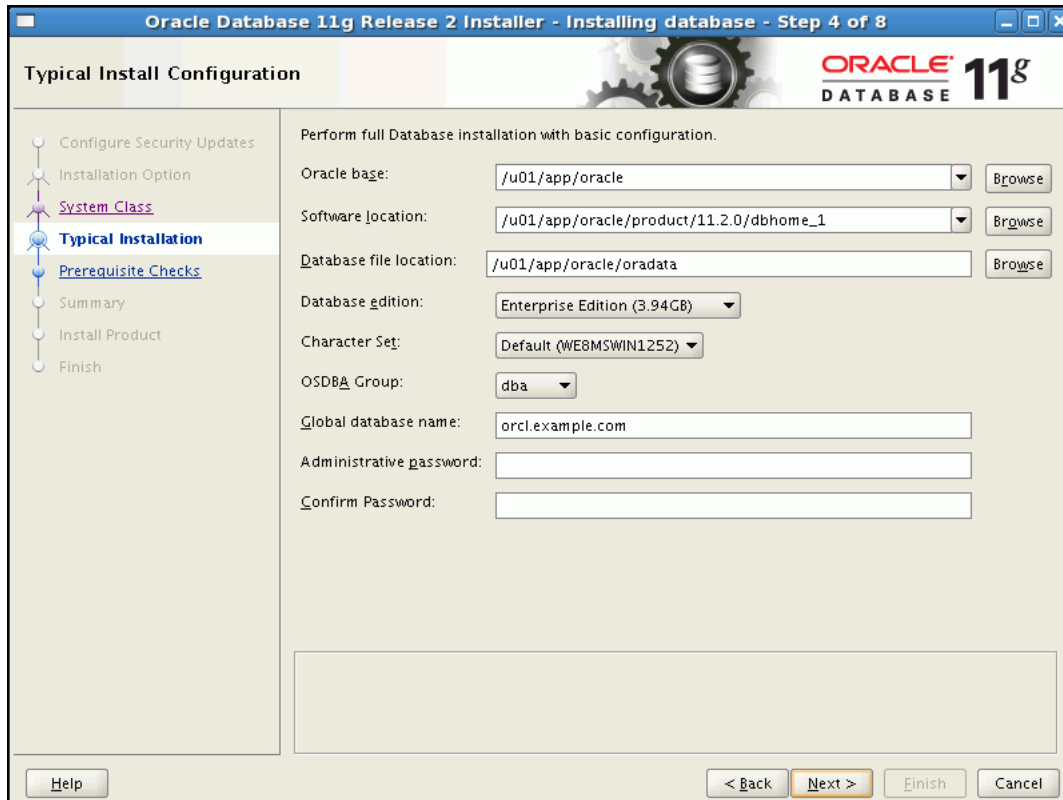
The System Class window appears.

5. Choose **Desktop Class**.

You can choose the Server Class option to customize your installation. For example, you use this method to configure Oracle Automatic Storage Management for your database, install the Sample Schemas, or configure automated backups. Selecting this option guides you through a series of installation steps that are not documented in this guide. For more information about the advanced choices, see "[About Advanced Installation](#)" on page 2-5. Also see *Oracle Database Installation Guide* for your platform.

Click **Next**.

The Typical Install Configuration window appears.



6. Provide the following configuration details for the database:

- Oracle Base Location— The Oracle base directory helps to facilitate the organization of multiple Oracle software installations. See the *Oracle Database Installation Guide* for your platform for more information about ORACLE_BASE.

If you did not set the ORACLE_BASE environment variable before starting OUI, then the Oracle base directory is created in an `app/username/` directory on the first existing and writable directory from `/u01` through `/u09` for UNIX and Linux systems, or on the disk drive with the most available space for Windows systems. If `/u01` through `/u09` does not exist on the UNIX or Linux system, then the default location is `user_home_directory/app/username`.

You can click **Browse** to find the directory you want to act as the Oracle base directory.

- Software Location—The software location is the Oracle home for your database. You must specify a new Oracle home directory for each new installation of Oracle Database software. By default, the Oracle home directory is a subdirectory of the Oracle base directory.

You can click **Browse** to find the directory where you want to install the Oracle Database software.

- Database File Location—The database file location is the location where Oracle Database files are stored. By default, this location is `Oracle_base/oradata`. You can click **Browse** to select a different location.

- Database Edition—Select either **Enterprise Edition**, **Standard Edition**, **Standard Edition One**, or **Personal Edition** (Microsoft Windows platforms only). See "[Installation Type](#)" on page 2-3.
- Character Set—Choose the character set to use to store the data within the database. You can choose between the **Default**, which is based on the operating system language settings, or **Unicode**.
- OSDBA Group (Linux and UNIX platforms only)—Specify the operating system DBA group. Host computer users in this group have administrative privileges on the database. This group is typically named **dba**. Refer to *Oracle Database Installation Guide for Linux* or for your UNIX platform for more details.
- Global Database Name—Enter the fully qualified global database name. See "[Installation Type](#)" on page 2-3 for more on global database name.
- Administrative Password—Specify the initial password for the **SYS**, **SYSTEM**, **SYSMAN**, and **DBSNMP** administrator accounts. If the password you choose is not a secure password, a warning message will be displayed.

After you enter the required information, click **Next**.

If Oracle software has not previously been installed on this server, then the Create Inventory window appears. If this is not the first installation attempt on this server, then the Perform Prerequisite Checks window appears.

7. *For first time installations on Linux and UNIX operating systems only*, specify a directory for installation files and the name of an operating system group that has write permissions for that directory.

If this is the first time you are installing any Oracle software on this computer, then the Create Inventory Directory window appears. You must specify a local directory for the inventory, which OUI uses to keep track of all Oracle software installed on the computer. This information is used while applying patches or upgrading an existing installation, and while deinstalling Oracle software. Note that this directory is different from the Oracle home directory. The recommended value for the inventory directory is *Oracle_base/./oraInventory*, or one level above the Oracle base directory, in the *oraInventory* subdirectory. If your Oracle base directory is */u01/app/oracle*, then the Oracle inventory directory defaults to */u01/app/oraInventory*.

In this window you can also specify the operating system group that has write permissions on the inventory directory. This prevents other users from writing over the Oracle product installation files. Typically the `oinstall` group is selected.

After you enter a directory path and optionally specify an operating system group, click **Next** to continue.

The Perform Prerequisite Checks window appears.

8. If any checks failed, then take corrective actions.

OUI performs several environment checks and indicates whether the check was a success, or resulted in a warning or failure. Details of the checks are provided in the displayed window. The installation can proceed only when all checks have a status of either Succeeded or Warning. If any of the environment checks failed, then they must be resolved manually. See "[Checking Prerequisites](#)" on page 2-2 for more information.

If all the prerequisite checks passed, or after you click **Next**, the Summary window appears,

9. Review the installation summary, then click **Finish** to start the installation.

The Install window appears, showing the installation progress. After the installation phase, the Configuration Assistants window appears. This window lists the configuration assistants that are started automatically. If you chose to create a starter database, then Database Configuration Assistant starts automatically in a separate window.

After database creation, a window is displayed that summarizes the database creation.

10. (Optional) Click **Password Management** to unlock user accounts to make the accounts accessible to users.

The `SYS` and `SYSTEM` accounts are unlocked by default.

11. Click **OK** to continue the installation.

12. *For Linux and UNIX operating systems only*, run the specified scripts, then click **OK**.

In the Execute Configuration Scripts window, 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**.

13. Make note of the information in the Finish window, then click **Close** to exit OUI.

Your installation and database creation is now complete. The Finish window displays one or more important URLs, including one for the Oracle Enterprise Manager Database Control console. (The console is the user interface—the Web application—portion of Oracle Enterprise Manager Database Control.)

You use Oracle Enterprise Manager Database Control to perform common database administration tasks. The URL and port information for Database Control can be found after installation in the `Oracle_home/install/portlist.ini` file.

You can access this URL with your browser and log in as the `SYS` user with the associated password, and connect as `SYSDBA`. See [Chapter 3, "Getting Started with Database Administration"](#) for more information about using Database Control.

Creating and Managing a Database with DBCA

Unless you specified that only the Oracle Database software should be installed, Oracle Universal Installer automatically runs Database Configuration Assistant (DBCA) after software installation is complete. DBCA then creates a database using the information you provided. If you do not create a starter database and later want to create one, or to create additional databases, use DBCA.

Note: With Oracle Database, you typically have a single database that hosts multiple applications. You do not need multiple databases to run different applications. Instead, you separate the objects that support each different application into different *schemas* in the same database. However, there may be situations in which you want to create multiple Oracle databases on the same host computer. When you do this with DBCA, the new databases typically use the same Oracle home directory as the first database, but store database data files separately from those of the first database.

DBCA also enables you to modify a database configuration, delete a database, and more. This section describes the following DBCA tasks:

- [Starting DBCA](#)
- [Creating a Database Using DBCA](#)
- [Changing the Configuration of a Database Using DBCA](#)
- [Deleting a Database Using DBCA](#)
- [Managing Templates with DBCA](#)

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

Starting DBCA

Follow the steps in this section to start DBCA.

Note: If you choose to create a starter database while installing the Oracle Database software, then OUI automatically starts DBCA.

To start DBCA:

1. Log on to your computer as a member of the administrative group that is authorized to install Oracle Database software and to create and run the database.
2. Do one of the following:
 - To start DBCA on a Microsoft Windows operating system, click **Start**, select **Programs** (or **All Programs**), then **Oracle - HOME_NAME**, then **Configuration and Migration Tools**, and then **Database Configuration Assistant**.
 - To start DBCA on UNIX or Linux, or at the command-line prompt on the Windows operating system, enter the following command:

```
dbca
```

The `dbca` utility is typically located in the `ORACLE_HOME/bin` directory.
3. Click **Next** to bypass the Welcome window.

Creating a Database Using DBCA

DBCA enables you to create an Oracle database by following a step-by-step guided workflow.

To create a database using DBCA:

1. Start DBCA as described in "[Starting DBCA](#)" on page 2-11.
2. In the Operations window, select **Create a Database** and click **Next** to start the guided workflow for creating a database. The workflow requests your input in the following windows:
 - [Step 2 - Database Templates](#)
 - [Step 3 - Database Identification](#)
 - [Step 4 - Management Options](#)

- [Step 5 - Database Credentials](#)
- [Step 6 - Database File Locations](#)
- [Step 7 - Recovery Configuration](#)
- [Step 8 - Database Content](#)
- [Step 9 - Initialization Parameters](#)
- [Step 10 - Database Storage](#)
- [Step 11 - Creation Options](#)

The following sections provide details on each window. Most windows provide default settings.

Step 2 - Database Templates

This window enables you to select the type of database to create. You can use DBCA to create a database from templates supplied by Oracle or from templates that you create. The templates contain settings optimized for a particular type of workload.

Oracle ships templates for the following two workload types:

- General purpose or transaction processing
- Data warehouse

Click **Show Details** to see the configuration for each type of database. Select the template suited to the type of workload your database will support. If you are not sure which to choose, then select the default **General Purpose or Transaction Processing** template.

For more complex environments, you can select the **Custom Database** option. This option does not use templates and results in a more extensive interview, which means that it takes longer to create your database.

For more information about using database templates, see "[Managing Templates with DBCA](#)" on page 2-19.

Step 3 - Database Identification

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

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

Step 4 - 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, and central management tools for managing your entire Oracle environment.

Select **Configure Enterprise Manager** to use Enterprise Manager. Then, select one option:

- If the Oracle Management Agent is installed on your host computer, then you can choose central management by selecting **Register with Grid Control for centralized management** and then selecting the Management Service.

- To manage your database locally, select **Configure Database Control for local management**. You can then optionally configure e-mail notification of database alerts or a daily backup of the database.

Step 5 - Database Credentials

In this window, specify the passwords for the administrative accounts such as `SYS` and `SYSTEM`.

Step 6 - Database File Locations

In this window, specify the type of storage you would like your database to use. For more information, see "[About Advanced Installation](#)" on page 2-5.

Then specify the locations for the Oracle database files. Select one of the following options:

- **Use Database File Locations from Template**—This option instructs DBCA to use the directory information as specified in the template. Later, you can make modifications to database file names 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 are created in this location. Later, you can make modifications to database file names and locations.
- **Use Oracle Managed Files**—This option instructs Oracle Database to directly manage operating system files comprising an Oracle database. You specify the default location, called a database area, for all your files. Oracle Database 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 online redo log files and control files, see [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 have to specify the file names, their location, or their sizes.

Step 7 - Recovery Configuration

When you create a new database, it is important to configure the database so you can recover your data if a system failure occurs. Online redo log files contain a record of changes that were made to data files. Online redo log files are stored in online redo log groups. You must have at least two online redo log groups for your database. After the online redo log files in a group have filled up, the log writer process (LGWR) switches the writing of redo records to a new online redo log group. Oracle Database can automatically save the inactive group of online redo log files to one or more offline destinations, known collectively as the **archived redo log** (also called the archive log). The process of turning online redo log files into archived redo log files is called **archiving**.

Archiving can be performed only if the database is running in `ARCHIVELOG` mode. A group of online redo log files cannot be reused by the log writer (LGWR) process until the group is archived. If the database is running in `NOARCHIVELOG` mode, then when a group becomes inactive after the LGWR process switches to a new group, the inactive group is available for immediate reuse by the LGWR process.

The `NOARCHIVELOG` mode protects a database from instance failure but not from media failure. Only the most recent changes made to the database, which are stored in the online redo log files, are available for instance recovery. To restore a database

operating in `NOARCHIVELOG` mode, you can use only entire database backups taken while the database is closed. Therefore, if you operate a database in `NOARCHIVELOG` mode, then back up the entire database at regular, frequent intervals.

The archiving of online redo log files has the following advantages:

- A database backup, with online and archived redo log files, guarantees that you can recover all committed transactions if the operating system or hardware fails.
- You can recover the database using a backup that was taken while the database was open and being used, if you have a copy of the archived log files that were written while the database was being backed up.
- You can perform online tablespace backups, and use these backups to restore a tablespace following media failure.
- You can keep a standby database current with its original database by continuously applying the original archived redo log files to the standby database.

Before you can archive the online redo log files, you must determine the destination to which you want to archive. Oracle recommends that the archive log be stored in a fast recovery area because it can simplify backup and recovery operations for your database. A **fast recovery area** is a location in which Oracle Database can store and manage files related to backup and recovery. It is distinct from the database area, which is a location for the current database files (data files, control files, and online redo log files).

When creating your database, you can select the following options:

- **Specify Fast Recovery Area**—Select this option to specify a backup and recovery area and its directory location and size. You can use variables to identify standard locations.
- **Enable Archiving**—Select this option to enable the archiving of database online redo log files, which can be used to recover a database. Selecting this option is the same as enabling Archive Log Mode in Oracle Enterprise Manager Database Control or running the database in `ARCHIVELOG` mode.

Oracle recommends you select **Enable Archiving**. Selecting this option provides better protection for your database for software or hardware failure. If you do not select this option now, then you can enable `ARCHIVELOG` mode later. See ["Configuring Your Database for Basic Backup and Recovery"](#) on page 9-5.

Step 8 - Database Content

When you create your database, you can load it with data using either of the following methods:

- **Sample Schemas**—Select **Sample Schemas** to include the Sample Schemas (`EXAMPLE`) tablespace in your database. Oracle guides and educational materials contain examples based upon the Sample Schemas. Oracle recommends that you include them in your database.
- **Custom Scripts**—In the **Custom Scripts** subpage, optionally specify one or more SQL scripts to run after your database is created. Scripts are useful for performing postinstallation tasks, such as loading custom schemas. Note that if you choose to run scripts after installation, then your scripts must include a connection string that identifies the database. Click **Help** for more information.

Step 9 - Initialization Parameters

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

- [Memory](#)
- [Sizing](#)
- [Character Sets](#)
- [Connection Mode](#)

You can also click the **All Initialization Parameters** button at the bottom of the window to display a list of all the database initialization parameters and their current settings.

Memory Use this window to set the initialization parameters that control how the database manages its memory. You can choose from the following methods for 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 DBCA assigns to the System Global Area (SGA) and the aggregate Program Global Area (PGA). With this setting, the system automatically tunes many of the memory components of the SGA, and allocates memory to individual PGAs as needed. The system can also dynamically decrease or increase the total amount of memory allocated to the SGA or aggregate PGA, depending on processing demands. The total memory used for the database instance never exceeds the amount you specify. This automatic memory tuning for the instance is known as **automatic memory management**. To learn more about PGA and SGA, see "[About Instance Memory Structure](#)" on page 5-5.
- **Custom**—This method requires more configuration than the Typical option, but gives you more control over how the database instance uses system memory. This option is meant for more experienced database administrators. You can directly specify memory sizes for the SGA and aggregate PGA and their substructures, such as the shared pool and buffer cache.

Select one of the following options for customized memory management:

- **Automatic Shared Memory Management** to allocate specific amounts of memory to the SGA and aggregate PGA. With this setting, automatic shared memory management is enabled for the SGA, and memory is allocated to the individual PGAs as needed.
- **Manual Shared Memory Management** to enter specific values for each SGA component and the aggregate PGA. This disables automatic shared memory management and enables you to determine how the SGA memory is distributed among the SGA memory components.

See Also:

- [Chapter 5, "Managing the Oracle Instance"](#) for more information about memory management options

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.

- **Block Size**—Use this list to select the block size, or accept the default. Oracle Database data is stored in data blocks of the size specified. One data block corresponds to a specific number of bytes of physical space on disk. Selecting a block size other than the default 8 kilobytes (KB) value requires advanced knowledge and should be done only when absolutely required.

While using predefined templates, this list is not enabled because the database will be created with the default block size of 8 KB. While using the custom option, you can change the block size.

- **Processes**—In this field, specify the maximum number of processes that can simultaneously connect to the database. Enter a number or accept the default value of 150. The default value for this parameter is appropriate for many environments. The value you select should allow for all background processes, user processes, and parallel execution processes.

Character Sets Use this tab to define the character sets used by your database.

Character sets are the encoding schemes used to display characters on your computer screen. The character set determines what languages can be represented in the database.

Oracle recommends using Unicode (AL32UTF8) as the database character set. AL32UTF8 is Oracle's name for the UTF-8 encoding of the Unicode standard. The Unicode standard is the universal character set that supports most of the currently spoken languages of the world. The use of the Unicode standard is indispensable for any multilingual technology, including database processing.

After a database is created and accumulates production data, changing the database character set is a time consuming and complex project. Therefore, it is very important to select the right character set at installation time. Even if the database does not currently store multilingual data but is expected to store multilingual data within a few years, the choice of AL32UTF8 for the database character set is usually the only good decision.

Even so, the default character set used by Oracle Universal Installer (OUI) and Database Configuration Assistant (DBCA) for the UNIX, Linux, and Microsoft Windows platforms is not AL32UTF8, but a Microsoft Windows character set known as an ANSI code page. The particular character set is selected based on the current language (locale) of the operating system session that started OUI or DBCA. If the language is American English or a Western European language, then the default character set is WE8MSWIN1252. Each Microsoft Windows ANSI Code Page can store data from only one language or a limited group of languages, such as only Western European, or only Eastern European, or only Japanese.

A Microsoft Windows character set is the default even for databases created on UNIX and Linux platforms because Microsoft Windows is the prevalent platform for client workstations. Oracle Client libraries automatically perform the necessary character set conversion between the database character set and the character sets used by non-Windows client applications.

You may also choose to use any other character set from the presented list of character sets. You can use this option to select a particular character set required by an application vendor, or choose a particular character set that is the common character set used by all clients connecting to this database.

Because AL32UTF8 is a multibyte character set, database operations on character data may be slightly slower when compared to single-byte database character sets, such as WE8MSWIN1252. Storage space requirements for text in most languages that use characters outside of the ASCII repertoire are higher in AL32UTF8 compared to legacy

character sets supporting the language. Note that the increase in storage space concerns only character data and only data that is not in English. The universality and flexibility of Unicode usually outweighs these additional costs.

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

Note: AL32UTF8 is a variable-width multibyte character set. Applications connecting to a database that uses AL32UTF8 for character data processing must be correctly programmed to work with such character sets. Always verify the character set requirements of the applications that use the database. Contact the application vendor and ask for a Unicode-capable version, if your current application version does not support the Unicode standard.

- **National Character Set**—In this list, select a character set or accept the default. The national character set, also called NCHAR character set, is the character set used to store and process data of data types NVARCHAR2, NCHAR, and NCLOB. These data types allow storing of Unicode characters in a database that does not have a Unicode database character set. Unless installation requirements of any of your applications specify otherwise, accept the default value of AL16UTF16 as the national character set.

Note: Although this character set is called "national," after the SQL standard (ISO/IEC 9075), it is not better suited to support globalized applications than the database character set. Because working with national character set data requires additional API calls in client applications, and because national character set data is not supported by some database components, such as Oracle Text, Oracle recommends that multilingual applications use VARCHAR2, CHAR, and CLOB data types and an Oracle database with the database character set AL32UTF8.

- **Default Language**—In this 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 reading direction (left to right or right to left).
- **Default Territory**—In this list, select the name of the territory whose conventions are to be followed for day and week numbering or accept the default. The default territory also establishes the default date format, the default decimal character and group separator, and the default International Standardization Organization (ISO) and local currency symbols. For example, in the United Kingdom, the default date format is DD-MON-YYYY, where DD is the day of the month (1-31), MON is the abbreviated name of the month, and YYYY is the 4-digit year.

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

- **Dedicated Server Mode**—This mode allows a dedicated server process for each user process. Select this option when the number of total clients is expected to be small, for example, 50 or fewer. You might also choose this option when database clients typically make persistent, long-running requests to the database. By default, the database is configured for dedicated server processes.
- **Shared Server Mode**—This mode allows several client connections to share a database-allocated pool of resources. Use this mode in configurations in which client load is expected to cause a strain on memory and other system resources. If you choose shared server mode, then you must indicate the number of server processes you want to create when a database instance is started. For more information about setting this parameter, click **Help**.

Step 10 - Database Storage

In this window, a navigation tree displays the storage structure of your database (control files, data files, online redo log groups, and so forth). If you are not satisfied with the storage structure or parameters, then you can make changes.

Note that if you selected a preconfigured template for a database, then you cannot add or remove control files, data files, or undo segments.

Step 11 - Creation Options

In this window, select any of the following options for creating the database:

- **Create Database**—Select this option to create your database now.
- **Save as a Database Template**—Select this option to save the database definition as a template to use at a later time.
- **Generate Database Creation Scripts**—Select this option to generate a SQL database creation script that you can run at a later time.

After you make your choice, click **Finish**. A confirmation window appears after a short period. Review the Create Database Summary information, then click **OK** to proceed.

Changing the Configuration of a Database Using DBCA

You can use DBCA to change the configuration of an existing database. For example, you can:

- Register the database with Grid Control
- Configure Database Control alert notification via e-mail
- Enable daily disk backups to the recovery area
- Change default security settings
- Change server mode from dedicated to shared, or the reverse
- Add database options that were not previously configured (for example, Oracle Label Security or Oracle OLAP)

To change the configuration of a database using DBCA:

1. Start DBCA as described in "[Starting DBCA](#)" on page 2-11.
2. In the Operations window, select **Configure Database Options** and click **Next**.

3. Follow the instructions in the DBCA guided workflow.

Deleting a Database Using DBCA

You can also use DBCA to delete a database. When DBCA deletes a database, it shuts down the database instance and then deletes all database files. On the Windows platform, it also deletes associated Windows services.

To delete a database using DBCA:

1. Start DBCA as described in "[Starting DBCA](#)" on page 2-11.
2. In the Operations window, select **Delete a Database** and click **Next**.
3. Select the database to delete and click **Finish**.

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 duplicate existing databases. The information in templates includes database options, initialization parameters, and storage attributes (for data files, tablespaces, control files, and online redo log files).

Templates can be used just like scripts, but they are more powerful than scripts because you have the option of duplicating a database. Duplication saves time because you copy the files of an existing database, referred to as a *seed database*, 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, then 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 computer to another.

Types of Templates

Templates are divided into the following types:

- Seed templates
- Nonseed templates

The characteristics of each are shown in [Table 2-1](#).

Table 2–1 DBCA Template Types

Type	File Extension	Includes Data Files	Database Structure
Seed	.dbc	Yes	<p>This type of template contains both the structure and the physical data files of an existing database, referred to as the <i>seed database</i>. Your new database starts as a copy of the seed database, and requires only the following changes:</p> <ul style="list-style-type: none"> ■ Name of the database ■ Destination of the data files ■ Number of control files ■ Number of online redo log groups ■ Initialization parameters <p>Other changes can be made after database creation using custom scripts that can be invoked by DBCA, command-line SQL statements, or Oracle Enterprise Manager Database Control.</p> <p>The data files for the seed database are stored in compressed RMAN backup format in a file with a .dfb extension. The seed database control file is stored in a file with .ctl extension. (This file is needed only when storing data files in Oracle Automatic Storage Management (Oracle ASM) disk groups or as Oracle Managed Files.) The .dbc file contains the location of the seed database data files and contains the source database name used to mount the control file.</p>
Nonseed	.dbt	No	<p>This type of template is used to create a new database. It contains the characteristics of the database to be created. Nonseed templates are more flexible than their seed counterparts because all data files and online redo log files are created to your specification, and names, sizes, and other attributes can be changed as required.</p>

DBCA Templates Provided by Oracle

Oracle provides the templates shown in [Table 2–2](#).

Table 2–2 Oracle-Provided DBCA Templates and Their Corresponding Workloads

Template	Workload
Data warehouse	<p>Users perform numerous, complex queries that process large volumes of data. Response time, accuracy, and availability are key issues.</p> <p>These queries (<i>SELECT</i> statements) range from a fetch of a few records to queries that sort thousands of records from many different tables.</p>
General Purpose or Transaction processing	<p>Many concurrent users perform numerous transactions that require rapid access to data. Availability, speed, concurrency, and recoverability are key issues.</p> <p>Transactions consist of reading (<i>SELECT</i> statements), writing (<i>INSERT</i> and <i>UPDATE</i> statements), and deleting (<i>DELETE</i> statements) data in database tables.</p>
Custom database	<p>This template allows you maximum flexibility in defining a database because you can change any of the settings for the database being created.</p>

Creating Templates Using DBCA

Follow the instructions in this section to create your own templates.

To create templates:

1. Start DBCA as described in "[Starting DBCA](#)" on page 2-11.
2. In the Operations window, select **Manage Templates** and click **Next**.
3. In the Template Management window, select **Create a database template** and one of the following options, and click **Next**.

- **From an existing template**

Using an existing template, you can create a new template based on the predefined 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, data files, and initialization parameters. User-defined schemas and their data *will not* be part of the created template. The source database can be either local or remote. Select 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 as well as data)**

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

When creating templates from existing databases, you can translate file paths into Optimal Flexible Architecture (OFA) or maintain existing file paths. OFA is a set of file naming and placement guidelines for Oracle software and databases. Using OFA is recommended if the target computer on which you plan to create a database using the template has a different directory structure than computer on which the template was defined. Standard file paths can be used if the target computer has a directory structure that is similar to the directory structure on the source computer.

4. Follow the instructions in the DBCA guided workflow to create your template.

Deleting Templates

When you delete a template, it is no longer available to create a new database or a new template.

To delete a template:

1. Start DBCA as described in "[Starting DBCA](#)" on page 2-11.
2. In the Operations window, select **Manage Templates** and click **Next**.
3. In the Template Management window, select **Delete a database template** and click **Next**.
4. Select the template to delete and click **Finish**.

Installation: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Installation OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/install/install.htm

Getting Started with Database Administration

This chapter provides a brief roadmap for administering your database. It introduces you to Oracle Enterprise Manager Database Control (Database Control), the Web-based interface for managing an Oracle database.

This chapter contains the following sections:

- [Managing Your Database: A Roadmap](#)
- [Introduction to Oracle Enterprise Manager Database Control](#)
- [Starting and Stopping the Database Control Console Process](#)
- [Accessing the Database Home Page](#)
- [Granting Access to Database Control for Nonadministrative Users](#)
- [Creating Database Control Administrative Users](#)
- [Setting Database Control Preferences](#)
- [Administering the Database with SQL-Based Management Tools](#)
- [Getting Started with Oracle Enterprise Manager: Oracle By Example Series](#)

Managing Your Database: A Roadmap

This section provides an overview of the tasks involved in managing an Oracle database instance. Each chapter in this guide describes a different task in detail.

To manage your Oracle database:

1. Start the database instance.

After the installation, your instance is 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 instance and later restart it.

See ["Shutting Down and Starting Up the Oracle Instance"](#) on page 5-9.
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 structures: tablespaces and data files, online redo log files, and control files. Create or modify storage structures as needed.

- See [Chapter 6, "Managing Database Storage Structures"](#).
4. Review memory allocation and adjust as needed.
See ["Managing Memory"](#) on page 5-15.
 5. Review, unlock, and reset passwords for predefined database users as needed. Create new users, and assign privileges and roles to them.
See [Chapter 7, "Administering User Accounts and Security"](#).
 6. Create the necessary schema objects, including tables, views, and indexes. Populate the tables with data.
See [Chapter 8, "Managing Schema Objects"](#).
 7. Create or review the backup strategy for the database and back up the database.
See [Chapter 9, "Performing Backup and Recovery"](#).
 8. Enable archiving of online redo log files, if not already done.
See ["Configuring Recovery Settings"](#) on page 9-7.
 9. Monitor database performance, diagnose performance problems, and tune the database as necessary.
See [Chapter 10, "Monitoring and Tuning the Database"](#).
 10. Investigate, gather diagnostic data for, and report to Oracle Support Services any critical database errors.
See [Chapter 11, "Investigating, Reporting, and Resolving Problems"](#).
 11. Keep Oracle Database software up-to-date with the latest releases.
See [Chapter 12, "Managing Oracle Database Software"](#).

Introduction to Oracle Enterprise Manager Database Control

Oracle Enterprise Manager Database Control (Database Control) is the primary tool for managing your Oracle database. It is installed with the database.

Using Database Control, you can perform administrative tasks such as creating schema objects (tables, views, indexes, and so on), managing user security, managing database memory and storage, backing up and recovering your database, and importing and exporting data. You can also view performance and status information about your database.

Database Home Page

The main page for database administration is the Database Home page. This is the page that loads when you log in to Database Control. See ["Accessing the Database Home Page"](#) on page 3-6.

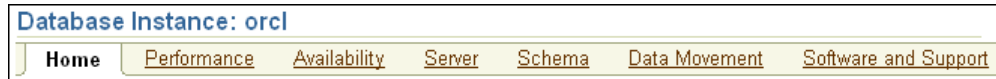
Online Help

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

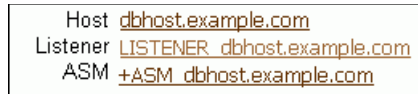
Navigation

Navigational features of Database Control include the following:

- Subpage links at the top of the page. These links take you to the various Database Control subpages that organize management tasks into distinct categories. For example, you click **Availability** to perform tasks associated with backing up and recovering your data.

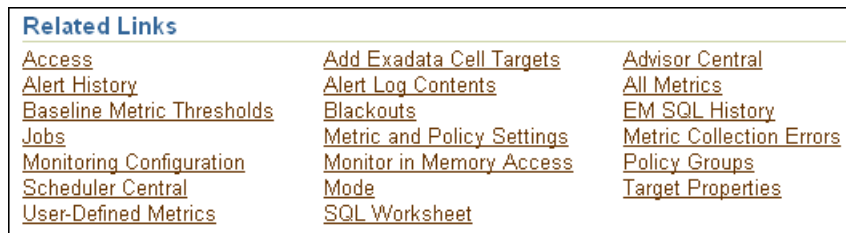


- Drill-down links that provide increasing levels of detail.



For example, clicking the link next to the label Host provides additional details about the computer on which the database runs.

- Suggested related links for more information.



Configuring the Operating System Environment Variables

Before using certain tools that access the Oracle database, such as SQL*Plus, you must configure environment variables for your operating system. These environment variables are used by Oracle Database to determine the database instance to which the tool should connect.

To configure operating system environment variables for your database instance on Linux and UNIX systems:

1. Open an operating system command window.
2. Ensure that the environment variables `ORACLE_HOME` and `ORACLE_SID` are set properly. The commands to use to set these environment variables depend on the shell you use to interface with the operating system. For example:
 - (bash or ksh) `export ORACLE_SID=orcl`
 - (csh or tcsh) `setenv ORACLE_SID=orcl`

You can set these with the scripts `coraenv` (for the C shell) and `oraenv` (for other shells). These scripts are typically located in the `/usr/local/bin` directory.
3. Ensure that the `$ORACLE_HOME/bin` directory is in your `PATH` environment variable.
4. You can also edit the profile file for your default shell in the home directory of the software owner, for example `/home/oracle`, so that these environment variables are set every time you log in as that user.

To configure operating system environment variables for your database instance on Windows systems:

1. Open an operating system command window.
2. Use either `regedit` or the Oracle Administration Assistant for Windows to make sure the `ORACLE_HOME` and `ORACLE_SID` parameters are set to the correct values in the `HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_HOME_NAME` registry subkey.
3. Ensure that the `%ORACLE_HOME%\bin` directory is in your `PATH` environment variable. At a command prompt, use a command similar to the following:

```
set PATH=%ORACLE_HOME%\bin;%PATH%
```

See Also:

- *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems* for more information about setting environment variables
- *Oracle Database Platform Guide for Microsoft Windows* for details on modifying the registry entries

Starting and Stopping the Database Control Console Process

To access Oracle Enterprise Manager Database Control (Database Control) from a browser, the Database Control console process `dbconsole` must be running on the database host computer. The `dbconsole` process is automatically started after installation. However, if the system must be restarted, you can start the `dbconsole` process manually at the command line or start it as a service in Windows. You can also run commands to stop the `dbconsole` process or view its status. If the `dbconsole` process is stopped, then it must be manually restarted before you use Database Control.

To start or stop, or to view the status of the dbconsole process from the command line:

1. Configure the operating system environment variables, as described in "[Configuring the Operating System Environment Variables](#)".
2. Do one of the following:
 - To start the `dbconsole` process, run the following command:

```
./emctl start dbconsole
```
 - To stop the `dbconsole` process, run the following command:

```
./emctl stop dbconsole
```
 - To view the status of the `dbconsole` process, run the following command:

```
./emctl status dbconsole
```

Note: If you are prompted to set the `ORACLE_UNQNAME` environment variable, then set this variable and run the `emctl` command again. Set the variable to the database unique name (the value of the `DB_UNIQUE_NAME` database parameter). A typical value for `DB_UNIQUE_NAME` is `orcl`.

To start or stop dbconsole as a service on Microsoft Windows:

1. Do one of the following:
 - On Windows XP and Windows Server 2003, click **Start**, and then select **Control Panel**.
 - On Windows 2000 Server, click **Start**, select **Settings**, and then select **Control Panel**.

The Control Panel window opens.

2. Double-click the **Administrative Tools** icon, and then double-click the **Services** icon.

The Services window opens.

Oracle Database services begin with `Oracle`. The `dbconsole` service is listed as `OracleDBConsoleOracle_sid`, where `Oracle_sid` is the system identifier (SID) for your database instance. The status of this process (either started or stopped) is listed in the Status column.

3. Double-click the `OracleDBConsoleOracle_sid` service.

The service properties window opens.

4. In the Startup Type list, ensure that either **Manual** or **Automatic** is selected.

5. Do one of the following:

- To start `dbconsole`, click **Start**, and then click **OK**.
- To stop `dbconsole`, click **Stop**, and then click **OK**.

Accessing the Database Home Page

The Database Home page is the main database management page in Oracle Enterprise Manager Database Control (Database Control).

To access the Database Home page:

1. Ensure that the `dbconsole` process is running on the database host computer.
See "[Starting and Stopping the Database Control Console Process](#)" on page 3-5.
2. In your Web browser, enter the following URL:

```
https://hostname:portnumber/em
```

For example, if you installed the database on a host computer named `comp42.example.com`, and the installer indicated that your Enterprise Manager Console HTTP port number is `1158`, then enter the following URL:

```
https://comp42.example.com:1158/em
```

If you have recently installed Oracle Database, you can determine the port number for Database Control on Linux and UNIX systems by viewing the

\$ORACLE_HOME/install/portlist.ini file. Otherwise, you can determine the port number for Database Control by searching for REPOSITORY_URL in the \$ORACLE_HOME/Oracle_sid/sysman/config/emd.properties file, where Oracle_sid is the system identifier (SID) for your database instance.

If the database instance is running, then the Login page appears when you access Database Control. If the database is down and must be restarted, Database Control displays the Startup/Shutdown and Perform Recovery page.

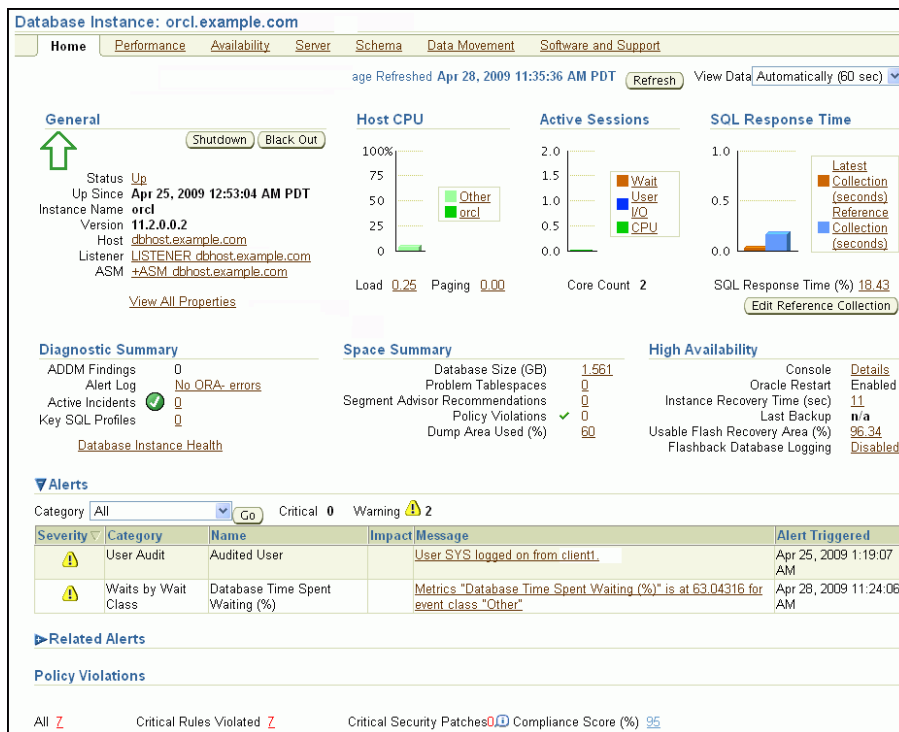
You must do the following to start the database:

- a. Click **Startup/Shutdown**, enter the host login user name and password, and then enter the database login user name and password.
For the database user name and password, enter *SYS* and the password that you specified during installation.
 - b. Click **OK** to start the database instance, and then in the Confirmation page, click **Yes** to start the database in open mode.
3. Log in to the database with a user account that is authorized to access Database Control.

This user initially could be *SYS* or *SYSTEM*, with the password that you specified during database installation.

Although the *SYSTEM* account can be used to perform day-to-day administrative tasks, Oracle strongly recommends creating a named user account for administering the Oracle database to enable monitoring of database activity. To back up, recover, or upgrade the database, you must log in as a *SYSDBA* user.

Database Control displays the Database Home page.



The subpage links across the top of the page enable you to access performance, availability, and other administration pages to manage your database. The features provided by these pages are discussed in other sections of this documentation.

The various sections of the Database Home page provide information about the environment and status of the database. For example, the 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 areas, and, in some cases, to obtain recommendations for resolving the problems. These topics are discussed in [Chapter 10, "Monitoring and Tuning the Database"](#).

See Also:

- ["Starting and Stopping the Database Control Console Process"](#) on page 3-5
- *Oracle Database 2 Day + Security Guide*

Granting Access to Database Control for Nonadministrative Users

As a database administrator, you can log in to Oracle Enterprise Manager Database Control (Database Control) with the `SYS` or `SYSTEM` user account to perform administrative and other tasks. Nonadministrative users may also want to log in to Database Control. For example, application developers may want to take advantage of the Database Control interface to create or modify tables, indexes, views, and so on. You must grant access to Database Control to these users before they can log in.

For nonadministrative users to have access to Database Control, they must be granted the `SELECT_CATALOG_ROLE` role. See ["Example: Granting Privileges and Roles to a User Account"](#) on page 7-16.

See Also:

- *Oracle Database 2 Day + Security Guide*

Creating Database Control Administrative Users

When you log in to Oracle Enterprise Manager Database Control (Database Control) using the `SYS`, `SYSTEM`, or `SYSMAN` user accounts, you are logging in as the Database Control super user. These are the only accounts that are automatically granted the roles and privileges required to administer Database Control itself. Examples of Database Control administration tasks include the following:

- Creating other Database Control administrators
- Configuring e-mail and other notification methods
See ["Setting Up Direct Alert Notification"](#) on page 10-14.
- Setting alert thresholds for database metrics, such as `tablespace space usage percentage exceeded` or `SQL response time exceeded`
See ["Managing Alerts"](#) on page 10-8.
- Selecting database policies to apply, so that Database Control can show alerts if a policy is violated. (An example policy is "A nonsystem user cannot use the `SYSTEM` or `SYSAUX` tablespace as its default tablespace.")
- Defining **blackouts**, which are time periods in which database monitoring is suspended so that maintenance operations do not skew monitoring data or generate needless alerts.
See ["Defining Blackout Periods"](#) on page 3-11.

You can create Database Control administrative users who have enough privileges to administer Database Control itself, but lack the high-level database administration privileges of the `SYS` and `SYSTEM` users. This practice enables you to assign the minimum privileges required for other Database Control administrators to do their jobs, which is a best practice for database security. You can also create a Database Control administrative account for yourself, thus avoiding logging in as `SYS` or `SYSTEM` until you must perform database administration tasks.

Using the following procedure, you can assign Database Control administrative privileges to an existing database user or create a new Database Control administrative user. When you create a new Database Control administrative user, a user account is created for that user for the database. You must then decide which system privileges, object privileges, or roles to grant the user, if any, to perform database administration tasks.

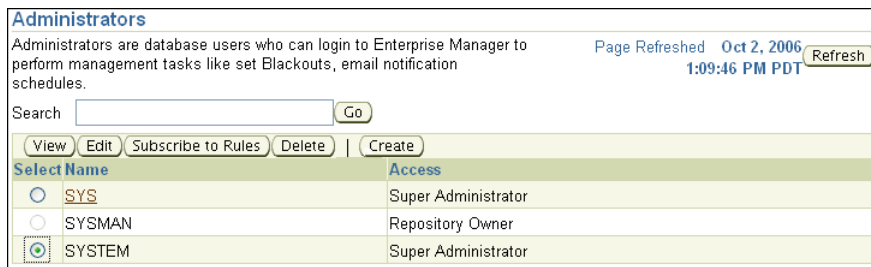
To create a Database Control administrative user:

1. On any Database Control page, at the top of the page, click **Setup**.

The Enterprise Manager Configuration page appears, showing the Overview of Setup page.

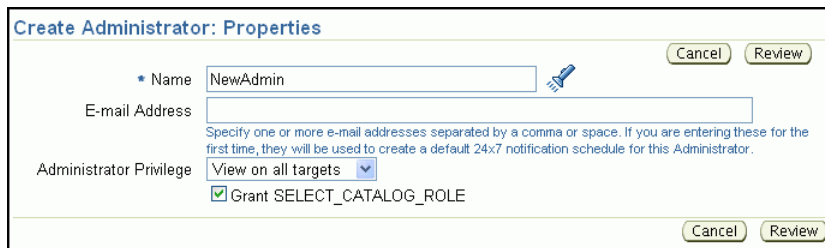
2. In the left navigation bar, click **Administrators**.

The Administrators page appears.



3. Click **Create**.

The Create Administrator: Properties page appears.



4. In the **Name** field, enter the name of an existing database user, or click the flashlight icon next to the field and select an existing database user.
5. Enter one or more e-mail addresses for this administrator only if you plan to set up e-mail notifications for the database.
See "[Setting Up Direct Alert Notification](#)" on page 10-14 for more information.
6. Click **Review** to view a page that summarizes the information that you entered.

7. Click **Finish** to configure the database user as a Database Control administrative user.

The Administrators page appears, showing the new administrator in the list.

Setting Database Control Preferences

This section discusses setting user preferences for Oracle Enterprise Manager Database Control (Database Control). It contains the following topics:

- [About Database Control Preferences](#)
- [Defining Blackout Periods](#)
- [Setting Preferred Credentials](#)

About Database Control Preferences

Database Control enables you to set user preferences in the following areas:

Notification

These settings enable Database Control to e-mail you alerts. Alerts are notifications that the database is in an undesirable state and needs your attention. By default, the Database Home page lists all alerts. However, setup is required for e-mail notification. For more information about alerts and setting up notifications, see "[Setting Up Direct Alert Notification](#)" on page 10-14.

Blackout Administration

Blackouts suspend the collection of database monitoring data and the sending of alerts by Database Control. Blackouts enable you to perform scheduled maintenance on the database without receiving needless alerts and without skewing the monitoring data.

For example, you can stop data collections during a database backup or a hardware upgrade. If you continue monitoring during these periods, then the collected data shows trends and other monitoring information that is not the result of typical day-to-day operations. To get a more accurate, long-term picture of database performance, you can use blackouts to exclude these special-case situations from data analysis. See "[Defining Blackout Periods](#)" on page 3-11 for more information.

Preferred Credentials

Database Control can automatically execute many routine administrative tasks, such as backups. You can use a job scheduling system that is built into Database Control for these routine tasks. To keep your environment secure, setting up tasks for automatic execution in Enterprise Manager requires you to provide login information for the host computer and database. To avoid entering this information every time you create or run a job or task, Database Control 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 3-13.

Defining Blackout Periods

You can define one or more one-time or repeating blackout periods during which collection of database monitoring information and the sending of alerts are suspended.

To define a blackout period:

1. On any Database Control page, at the top of the page, click **Setup**.

The Enterprise Manager Configuration page appears, showing the Overview of Setup page.

- In the left-hand pane, click **Blackouts**.

The Blackouts page appears.

- Click **Create** to start the Create Blackout wizard.

The Create Blackout: Properties page appears.

- (Optional) Replace the default blackout name with one of your choosing.
- (Optional) In the **Comments** field, enter text that describes the purpose of the blackout.
- In the **Reason** list, select the blackout reason that is most appropriate.
- In the Available Targets section, in the Type list, select **Database Instance**.
The fully qualified name for your database instance appears in the Available Targets list.
- In the Available Targets list, select your instance, and then click the **Move** icon.

Note: You can also double-click the instance name.

The instance name is moved from the Available Targets list to the Selected Targets list.

- Click **Next**.

The Create Blackout Schedule page appears.

10. On the Create Blackout Schedule page, do the following:
 - a. In the Start section, schedule the blackout either immediately or for a later date and time.
 - b. In the Duration section, indicate the duration of the blackout.
 - c. To repeat the blackout periodically, in the Repeating section, select a repeat frequency from the Repeat list. Otherwise, use the default value of Do Not Repeat.

11. Click **Next**.

The review page appears.

Review what you have entered. You can click **Back** to change a setting.

12. Click **Finish**.

The Confirmation heading appears on the Blackouts page, and the new blackout period is shown in the list.

See Also:

- ["About Alerts"](#) on page 10-1





Setting Preferred Credentials

When you set preferred credentials, Database Control automatically fills in host computer and database login credentials for you at times when it usually prompts for these credentials. Database Control also fills in these credentials when it is about to run a job that requires credentials.

To set preferred credentials for the database:

1. From any Database Control page, at the top of the page, click **Preferences**.
The Preferences page appears.
2. Click **Preferred Credentials** in the left-hand pane.

The Preferred Credentials page appears, showing a table of targets.

Preferred Credentials			
You can use preferred credentials to simplify access to targets that you manage. If a target has preferred credentials set, applications that log in to that target will automatically use the preferred credentials.			
Target Type	Total Targets	Targets with Credentials Set	Set Credentials
Listener	1	0	
Host	1	0	
Database Instance	1	0	
Agent	1	0	

- In the table row for the Database Instance, click the icon under **Set Credentials**.

The Database Preferred Credentials page appears.

- Enter the following credentials:
 - Database credentials for **Username/Password** and **SYSDBA Username/SYSDBA Password**
 - Host credentials for **Host Username/Host Password**

For example, you might typically connect as the user `SYSTEM`, use the `SYS` account for `SYSDBA` access, and use `oracle` for the host user name.

Note: The host user may require certain host privileges to run background jobs such as database backups. For example, on UNIX and Linux, the host user must belong to the `OSDBA` group (typically `dba`). On Windows, the host user must be a member of the `Administrators` group and must be granted the `Log on as batch job` logon right. See your platform documentation for more information.

- Click **Test** to test your credentials.
A confirmation message is displayed if your credentials can be verified.
- Click **Apply** to apply the changes.

Administering the Database with SQL-Based Management Tools

In addition to using the graphical user interface (GUI) pages presented in Database Control, you can use other Oracle tools such as SQL Developer and SQL*Plus to administer your database. These tools enable you to perform database management operations, and to query, insert, update, or delete data directly in the database.

The following sections provide details:

- [About SQL](#)
- [About SQL*Plus](#)
- [Starting SQL*Plus and Connecting to the Database](#)
- [About SQL Developer](#)

See Also:

- Oracle Database 2 Day Developer's Guide*
- Oracle Database SQL Language Reference*

About SQL

To perform many of its operations, Oracle Enterprise Manager Database Control (Database Control) submits structured query language (SQL) statements to the database. SQL (pronounced like *sequel*) is an industry-standard English-like computer programming language for querying and updating databases.

The following is an example of a SQL query that lists information about countries in a `COUNTRIES` table, which is owned by user `hr`:

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

SQL is a powerful language that can also be used to perform a variety of database administration tasks. The following SQL statement creates the database user `nick` and assigns him a password of your choosing, represented by *password*:

```
CREATE USER nick IDENTIFIED BY password;
```

When performing an administrative task in Database Control, you can click **Show SQL** to see the SQL statements that Database Control generates and submits.

About SQL*Plus

SQL*Plus is a command-line program that you use to submit SQL and PL/SQL statements to an Oracle database. You can submit statements interactively or as SQL*Plus scripts. SQL*Plus is installed with the database and is located in your `ORACLE_HOME/bin` directory.

You can start SQL*Plus from the command line, or on Microsoft Windows, from the Start menu.

When SQL*Plus loads, it issues the SQL prompt, which looks like this:

```
SQL>
```

At the SQL prompt, you can enter statements that perform administrative tasks such as shutting down the database or creating a new user, or you can query, insert, update, and delete data.

You can enter a single SQL statement on multiple lines. You must end each statement with a semicolon (;). For most statements, you can rerun a statement by entering a slash (/) on a line by itself.

See Also:

- *SQL*Plus User's Guide and Reference*
- *Oracle Database Concepts*

Starting SQL*Plus and Connecting to the Database

The section describes how to start SQL*Plus and connect to the database from both the command line and the Windows start menu.

For a new installation, you connect to the database using either the `SYS` or `SYSTEM` database accounts. When you enter `SYS` or a slash (/) as the user name and provide the `AS SYSDBA` clause, your access is authenticated using operating system authentication. **Operating system authentication** uses your Windows, UNIX, or Linux host user account to authenticate you to Oracle Database. You must have logged in to the host computer with a user account that is a member of a special host user group. On UNIX and Linux, this user group is typically `dba`. This type of authentication

enables you to connect to an Oracle database that is not yet started, so that you can start it up. See *Oracle Database Administrator's Guide* for more information.

The following procedures show how to log in to the database as user `SYS` using the `SYSDBA` privilege.

To start SQL*Plus and connect to the database from the command line:

1. Open a command window.
2. Configure the operating system environment variables, as described in ["Configuring the Operating System Environment Variables"](#).
3. Start SQL*plus using a command in the following format:

```
sqlplus {username | /} [as sysdba]
```

An example of this command is the following:

```
$ sqlplus / AS SYSDBA
Enter password: password
```

For *username*, you can use the `SYS` or `SYSTEM` administrative users. At the prompt, enter the password that you set up during installation. If you use the `SYS` user, you must include `AS SYSDBA` after the username.

SQL*Plus connects you to the default database instance (Microsoft Windows) or the database instance specified by environment variables (Linux and UNIX).

To start SQL*Plus and connect to the database from the Windows Start menu:

1. Configure the operating system environment variables, as described in ["Configuring the Operating System Environment Variables"](#).
2. Click **Start**, select **Programs** (or **All Programs**), then **Oracle - HOME_NAME**, then **Application Development**, and then **SQL*Plus**.
3. When prompted, enter the **user name** and **password** for the account to use to connect to the database.

For the user name, you can use the `SYS` or `SYSTEM` administrative accounts, and you can use the password that you set up during installation.

If you use `SYS` or `/` as the user name, follow them with a space and then the clause `AS SYSDBA`, as shown in the following examples:

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

or

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

See Also:

- ["Connecting to an Oracle Database from a Client Computer"](#) on page 4-7
- ["About Administrative Accounts and Privileges"](#) on page 7-4
- *Oracle Database Administrator's Guide*
- *SQL*Plus User's Guide and Reference*
- *Oracle Database SQL Language Reference*

About SQL Developer

SQL Developer provides another GUI for accessing your Oracle database. SQL Developer supports development in both the SQL and PL/SQL languages. It is available in the default installation of Oracle Database.

With SQL Developer, you can browse database objects, run SQL statements and SQL scripts, and edit and debug PL/SQL statements. You can also run any number of provided reports, also create and save your own.

You can download SQL Developer from the Oracle Technology Network (OTN) Web site.

See Also:

- *Oracle Database 2 Day Developer's Guide* for instructions for starting SQL Developer
- *Oracle Database Concepts*

Getting Started with Oracle Enterprise Manager: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Getting Started with Oracle Enterprise Manager OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/gettingstarted/gettingstarted.htm

Configuring the Network Environment

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

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

- [Understanding Network Configuration](#)
- [Viewing Listener Configuration](#)
- [Starting and Stopping the Listener](#)
- [Connecting to an Oracle Database from a Client Computer](#)
- [Networking: Oracle By Example Series](#)

Understanding Network Configuration

A **client** is any application that connects to Oracle Database to send or retrieve data. An Oracle Database client application can reside on any computer that has Oracle Database client software installed.

Oracle Net is a software layer that resides on the client computer and on the Oracle Database host computer. It establishes and maintains the connection between the client application and the database over a network, and exchanges messages between them using industry standard protocols.

For a client application and a database to communicate, the client application must be able to identify the database it wants to connect to, and the database must provide an identification. You can use a service name to connect to a database. A **service name** is a logical representation of a database, which is the way a database is presented to clients. A single database can be presented as multiple services.

Service names can provide location transparency so that the client application does not have to know the server's location. If the database is moved to another location, then you must reconfigure only Oracle Net. No changes are necessary to client applications.

This section contains these topics:

- [Oracle Net Listener Configuration](#)
- [Client Connections](#)
- [Tools for Network Configuration](#)

Oracle Net Listener Configuration

On the database host, the Oracle Net listener (the listener), is a process that listens for client connection requests. It receives incoming client connection requests and manages the traffic of these requests to the database server.

The default listener configuration file is called `listener.ora`, and it is located in the `network/admin` subdirectory of the Oracle home directory. For example, if your Oracle home directory is `/u01/app/oracle/product/11.2.0/dbhome_1`, then the `listener.ora` file is created by default in the `/u01/app/oracle/product/11.2.0/dbhome_1/network/admin` directory.

The file contains a protocol address that identifies the database. This address defines the protocol the listener is listening 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=dbhost.example.com)(PORT=1521)))
```

This example shows a TCP/IP protocol address that specifies the host computer of the listener and a port number. The listener can listen for connection requests on network interfaces with either IP version 4 (IPv4) or IP version 6 (IPv6) addresses.

The `listener.ora` file is automatically configured during installation, but you can reconfigure it with Oracle Enterprise Manager Database Control.

Because the configuration parameters have default values, you can start and use a listener without configuring it. This default listener is named `LISTENER`, supports no service names on startup, and listens on the following TCP/IP protocol address:

```
(ADDRESS=(PROTOCOL=tcp)(HOST=host_name)(PORT=1521))
```

An Oracle database registers with the listener within a minute or so of starting up. The service names, or the databases that they represent, to which the listener forwards client requests, can be configured in the `listener.ora` file. This information can also be dynamically registered with the listener. Dynamic registration of services and databases with the listener is called **service registration**.

Service registration is performed by the process monitor (PMON) process—an instance background process—of each database instance. Dynamic service registration does not require modification of the `listener.ora` file.

See Also:

- *Oracle Database Concepts* for more information about listeners and service names
- *Oracle Database Net Services Administrator's Guide* for more information about configuring listeners
- ["Viewing Listener Configuration"](#) on page 4-5
- ["Starting and Stopping the Listener"](#) on page 4-6
- ["About Background Processes"](#) on page 5-3 for more information about database processes

Client Connections

The following sections describe the elements involved in client connections to the database:

- [Connect Descriptors](#)

- [Connection Requests](#)
- [Naming Methods](#)

Connect Descriptors

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

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

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

Connection Requests

Users initiate a connection request by providing a connect string. A connect string includes a user name and password, and a connect identifier. This connect identifier can be the connect descriptor itself, or a name that resolves to the connect descriptor using mapping information stored in one or more repositories accessed with the naming methods described in "[Naming Methods](#)" on page 4-4. This name is referred to as a **net service name**.

Naming Methods

A **naming method** is a resolution method used by a client application to resolve a connect identifier to a connect descriptor when attempting to connect to a database service.

Oracle Net provides support for the following naming methods:

- **Easy Connect Naming**

The easy connect naming method enables clients to connect to an Oracle database by using only a TCP/IP connect string consisting of a host name and service name. The easy connect naming method requires no configuration. See "[Connecting to an Oracle Database from a Client Computer](#)" on page 4-7 for an example of easy connect naming.

- **Local Naming**

The local naming method stores connect descriptors, identified by their net service names, in a client configuration file named `tnsnames.ora`. This file is located in the `ORACLE_HOME/network/admin` directory. When you create a database using Oracle Database Configuration Assistant (DBCA), local naming is configured automatically. You must then use the Net Configuration Assistant to create connect descriptors and their corresponding net service names.

- **Directory Naming**

Directory naming resolves a database service, net service name, or net service alias to a connect descriptor stored in an LDAP-compliant directory server.

See Also:

- *Oracle Database Net Services Administrator's Guide*
- *Oracle Database Administrator's Guide* for more information about database services

Tools for Network Configuration

Oracle Database enables you to manage your network configuration with the following tools:

- [Net Configuration Assistant](#)
- [Oracle Enterprise Manager](#)
- [Oracle Net Manager](#)

Net Configuration Assistant

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

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

Oracle Enterprise Manager

Oracle Enterprise Manager enables you to manage your server-side network configuration with a Listener page. The Listener page displays information such as the current status, the Oracle home directory in which the listener configuration files are located, and the host name.

Oracle Net Manager

Oracle Net Manager provides the same features as Oracle Enterprise Manager with the addition of configuring profiles.

See Also:

- ["Viewing Listener Configuration"](#) on page 4-5
- *Oracle Database Net Services Administrator's Guide*

Viewing Listener Configuration

The Oracle Net listener (the listener) runs on your database host and handles incoming client requests. With Oracle Enterprise Manager Database Control (Database Control), you can view the status of the listener, which is set to start automatically whenever your host is restarted. You can also view the listener status at the command line.

To view information about the listener with Database Control:

- On the Database Home page, under the heading General, click the **Listener** link.

The Listener page appears, displaying the following information about the listener:

- Current status

- Availability over the last 24 hours
- Alias
- Version
- Oracle home for the listener
- Net address
- Configuration file (`listener.ora`) location
- Start time
- Host
- (Optional) Status of Oracle Restart

To view information about the listener at the command line:

1. Open a command window.
2. Follow the steps listed in "[Configuring the Operating System Environment Variables](#)".
3. Enter the following command:

```
lsnrctl status
```

Starting and Stopping the Listener

The Oracle listener is set to start automatically whenever the host is restarted. However, when your system encounters unforeseen circumstances, or when you have manually stopped the listener, you can restart it with Database Control or at the command line.

To stop the listener with Database Control:

1. On the Database Home page, under General, click the **Listener** link.
The Listener page appears.
2. Click **Stop**.
3. Enter the host credentials, if you did not configure Preferred Credentials, then click **Login**.
4. On the Start/Stop Listener page, click **OK** to perform the specified action or **Cancel** to quit.

To start the listener with Database Control:

1. On the Database Home page, under the heading General, click the **Listener** link.
The Listener page appears.
2. Click **Start**.
3. On the Start/Stop Listener page, click **OK**.

After a couple of minutes, a confirmation message appears, indicating that the listener has been started.

Note: If you have just stopped the listener and are still at the Database Control Listener page, then you will not see a Start button. You must go to the Database Home page, wait a minute or two, and then return to the Listener page.

To start or stop the listener at the command line:

1. Open a command window.
2. Follow the steps listed in "[Configuring the Operating System Environment Variables](#)".
3. Enter either of the following commands, depending on whether you want to start or stop the listener:

```
lsnrctl start
lsnrctl stop
```

Connecting to an Oracle Database from a Client Computer

This section describes how to use SQL*Plus and the easy connect naming method to connect to an Oracle database from a client computer. SQL*Plus is typically installed when you install Oracle Database client software. The easy connect naming method provides TCP/IP connectivity to databases without requiring you to configure Oracle Net Services.

You can use the instructions in this section to test your network configuration.

To connect to an Oracle database from a client computer using easy connect naming:

1. Complete the steps in "[Configuring the Operating System Environment Variables](#)" on page 3-4.
2. Do one of the following to start SQL*Plus:
 - (UNIX, Linux, or Windows systems) Open a command window and enter the following command:
3. When prompted, enter the user name followed by an at sign (@) and a connect identifier, where the connect identifier has the following format:

```
"host[:port][:/service_name][:server][/instance_name]"
```

The place holders used in the connect identifier format represent:

- *host* — the name or IP address of the Oracle database host computer.
Both IPv4 and IPv6 addresses are supported. IPv6 addresses must be enclosed in square brackets. See *Oracle Database Net Services Administrator's Guide* for information about IPv6 addressing.
- *port* (optional) — the TCP port number on which the Oracle Net listener listens for connections.

If *port* is excluded, then the standard port number 1521 is assumed.

- *service_name* — a database service name.

If no database service names are defined, then you can use the name of the default service that is created for the database. This service name consists of the global database name, which is made up of the `DB_NAME` and `DB_DOMAIN` initialization parameters as follows:

```
DB_NAME.DB_DOMAIN
```

If `DB_DOMAIN` is null, then the standard service name is just `DB_NAME`.

- *server* — the type of service handler. Acceptable values are `dedicated`, `shared`, and `pooled`. If omitted, then the default type of server is chosen by the listener: `shared` server if configured, otherwise `dedicated` server.
- *instance_name* — the instance to which to connect. When you specify only instance name, you connect to the default database service. If there is no default service configured in the `listener.ora` file, then an error is generated. You can obtain the instance name from the `INSTANCE_NAME` initialization parameter.

For example, to connect as user `NICK` to the database service `orcl.example.com` on the host `dbhost.example.com`, enter the following at the user name prompt:

```
nick@"dbhost.example.com/orcl.example.com"
```

The following examples substitute IPv4 and IPv6 addresses for the host name:

```
nick@"192.0.2.1/orcl.example.com"
```

```
nick@"[2001:0DB8:0:0::200C:417A]/orcl.example.com"
```

4. When prompted, enter the user password.

See Also:

- *Oracle Database Administrator's Guide* for more examples of connecting with SQL*Plus and for a discussion of environment variables
- *Oracle Database Net Services Administrator's Guide* for information about easy connect, connect identifiers, and other naming methods
- *Oracle Database Net Services Reference* for information about how to define the default service in `listener.ora`
- *SQL*Plus User's Guide and Reference*

Networking: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Networking OBE, enter the following URL in your browser:

```
http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day\_dba/network/network.htm
```

Managing the Oracle Instance

This chapter provides background information about the Oracle instance and instructions for managing the instance.

This chapter contains the following sections:

- [Overview of the Oracle Instance and Instance Management](#)
- [Shutting Down and Starting Up the Oracle Instance](#)
- [Viewing and Modifying Initialization Parameters](#)
- [Managing Memory](#)
- [Instances: Oracle By Example Series](#)

Overview of the Oracle Instance and Instance Management

An Oracle database system consists of an Oracle database and an Oracle instance.

A **database** consists of a set of disk files that store user data and metadata. **Metadata**, or "data about the data," consists of structural, configuration, and control information about the database.

An **Oracle instance** (also known as a **database instance**) contains the set of Oracle Database background processes that operate on the stored data and the shared allocated memory that those processes use to do their work.

Each instance has an instance ID, also known as a system ID (SID). Because there can be multiple Oracle databases on a host computer, each with its own set of data files, you must identify the instance to which you want to connect. For a local connection, you identify the instance by setting operating system environment variables `ORACLE_SID` and `ORACLE_HOME`. For a remote connection, you identify the instance by specifying a network address and a database service name.

An Oracle instance must be started to read and write information to the database. The Oracle instance creates the database upon receipt of instructions from the Oracle Database Configuration Assistant (DBCA) utility or the `CREATE DATABASE SQL` statement.

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

The properties of an Oracle 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 concepts of an Oracle instance and its management. It contains the following topics:

- [About Initialization Parameters](#)
- [About Background Processes](#)
- [About Server and Client Processes](#)
- [About Instance Memory Structure](#)
- [About Instance Startup and Shutdown](#)

See Also:

- *Oracle Database Concepts* for an overview of the Oracle database instance

About Initialization Parameters

Managing an Oracle instance includes configuring parameters that affect the basic operation of the Oracle instance. These parameters are called initialization parameters. The Oracle instance reads initialization parameters from a file at startup.

During installation, when you select a preconfigured database workload available in DBCA, the initialization parameters are optimized for typical use in the environment that you specified. As the number of database users increases and the workload increases, you might have to alter some initialization parameters. You can make these changes using the Initialization Parameter page in Database Control, or by using an advisor provided by Oracle Database, such as the Memory Advisor. See "[Optimizing Memory Usage with the Memory Advisors](#)" on page 10-38 for more information.

After being read from a file, initialization parameters are retained in memory, where the values for many of them can be changed dynamically. There are two types of parameter files. The type of file used to start the instance determines if dynamic initialization parameter changes persist across database shutdown and startup. The parameter file types are:

- 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. It *must not* be edited manually. It is stored on the host computer on which Oracle Database is running. Changes are made when you use Database Control to modify one or more initialization parameters, or when Oracle Database itself makes changes for self-tuning purposes. Any changes to it persist across database shutdown and startup operations.

Note: When changing an initialization parameter in the server parameter file, you can also specify that the in-memory value be changed, so that your change is reflected immediately in the current instance. If you do not change the in-memory value, then the change does not take effect until you shut down and restart the database.

- Text initialization parameter file

A **text initialization parameter file** is a text file that can be read by the Oracle instance, but it is not written to by the instance. You can change a text initialization parameter file with a text editor, but changes do not take effect until you restart the Oracle instance. When you start the instance with this type of file, you can still change many initialization parameters dynamically with Database Control, but only for the current instance. Unless you also edit the text initialization parameter

file and make the same change, the change is lost when you restart the database instance.

You can use SQL statements to create the following:

- A server parameter file from a text initialization file
- A server parameter file from the current (in-memory) values of all initialization parameters
- A text initialization parameter file from a server parameter file

When you create the database with DBCA, a server parameter file is created. This file is then used each time the database is started.

See Also:

- ["Viewing and Modifying Initialization Parameters"](#) on page 5-12
- *Oracle Database Administrator's Guide* for information about the default name and location of the server parameter file, and for commands to create a server parameter file or text initialization parameter file
- *Oracle Database Concepts* for an overview of parameter files

About Background Processes

The background processes of the Oracle instance manage memory structures, asynchronously perform I/O to write data to a file on a disk, and perform general maintenance tasks. The background processes consolidate functions that would otherwise be handled by multiple Oracle Database programs running for each user process. They monitor other Oracle Database 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. Some fundamental background processes are described in [Table 5-1](#).

Table 5-1 Oracle Database Background Processes

Background Process	Description
Database writer (DBWn)	The database writer writes modified blocks from the database buffer cache to the files on a disk. Oracle Database allows a maximum of 36 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 System Global Area (SGA) and the log writer process writes the redo log entries sequentially into an online redo log file.
Checkpoint (CKPT)	At specific times, all modified database buffers in the SGA are written to the data files by a database writer process (DBWn). This event is called a checkpoint . The checkpoint process signals DBWn, updates the data files and control files of the database, and records the time of this update.
System monitor (SMON)	The system monitor performs instance recovery when a failed instance is restarted.
Process monitor (PMON)	The process monitor performs a recovery when a user process fails. It cleans up the cache and frees resources that the failed process was using.

Table 5–1 (Cont.) Oracle Database Background Processes

Background Process	Description
Archiver (ARC <i>n</i>)	Archiver processes copy the online 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 processes. For more information, see Chapter 9, "Performing Backup and Recovery" .
Manageability monitor (MMON)	This process performs various management-related background tasks, for example: <ul style="list-style-type: none"> ▪ Issuing alerts whenever a given metric violates its threshold value ▪ Taking snapshots by spawning additional processes ▪ Capturing statistical values for SQL objects that have been recently modified
Job Queue Processes (CJQ0 and Jnnn)	Job queue processes run user jobs, often in batch mode. A job is a user-defined task scheduled to run one or more times.

See Also:

- *Oracle Database Reference* for a complete list of Oracle Database background processes
- *Oracle Database Concepts* for an overview of background processes

About Server and Client Processes

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

- A client program acting on behalf of the user, such as Oracle Enterprise Manager (Enterprise Manager), SQL*Plus, or an application
- A server process that handles the connection to the database on behalf of the client program, and that performs much of the work for the client program, such as parsing and running SQL statements, and retrieving and returning results to the client program

Server processes can be either dedicated or shared. When server processes are dedicated, Oracle Database is running in **dedicated server mode**. When server processes are shared, Oracle Database is running in **shared server mode**. In dedicated server mode, each client process has its own server process. Although 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 user connection, requires less memory for each user connection, and enables more users to access the database. Shared server mode is more efficient at supporting multiple client programs making frequent short-running queries.

About Instance Memory Structure

The sizes of the instance memory structures affect database performance and are controlled by initialization parameters.

Upon installation, you can let the database manage memory for you automatically, or you can choose to manually configure the instance memory structures. If you choose manual memory management, then Oracle Database provides advisors to help you determine appropriate values for memory parameters. For both manual and automatic memory management, Oracle Database sends alerts that identify memory sizing problems that require your attention.

The following sections describe the two important memory areas that you can monitor and size:

- [System Global Area](#)
- [Program Global Area](#)

See Also:

- ["Managing Memory"](#) on page 5-15
- *Oracle Database Concepts* for more information about Oracle Database memory architecture

System Global Area

The System Global Area (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 a physical disk.

[Table 5–2](#) lists the components of the SGA.

Table 5–2 SGA Components

Component	Description
Database buffer cache	Before data stored in the database can be queried or modified, it must be read from a disk and stored in the buffer cache. All user processes connected to the database share access to the buffer cache. For optimal performance, the buffer cache should be large enough to avoid frequent disk I/O operations.
Shared pool	The shared pool caches information that is shared among users: <ul style="list-style-type: none"> ■ 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 are executable code that is stored in the database
Redo log buffer	This buffer improves performance by caching redo information until it can be written to the physical online redo log files stored on disk. Redo information and online redo log files are discussed in "About Online Redo Log Files" on page 6-4.
Large pool	This optional area is used to buffer large I/O requests for various server processes.
Java pool	The Java pool is an area of memory that is used for all session-specific Java code and data within the Java Virtual Machine (JVM).
Streams pool	The Streams pool is an area of memory that is used by the Oracle Streams feature. For more information about Oracle Streams, see <i>Oracle Streams Concepts and Administration</i> .

Table 5–2 (Cont.) SGA Components

Component	Description
Result cache	The result cache buffers query results. If a query is run for which the results are stored in the result cache, then the database returns the query results from the result cache instead of rerunning the query. This SGA component speeds the execution of frequently run queries.

Program Global Area

A **Program Global Area (PGA)** is a memory area used by a single Oracle Database server process. When you connect to an Oracle database instance, you create a **session** that uses a server process for communication between the client and database instance. Each server process has its own PGA.

The PGA is used to process SQL statements and to hold logon and other session information. A large part of the PGA is dedicated to **SQL work areas**, which are working memory areas for sorts and other SQL operations.

The amount of PGA memory used and the contents of the PGA depend on whether the instance is running in dedicated server or shared server mode.

The total memory used by all individual PGAs is known as the **total instance PGA memory**, and the collection of individual PGAs is referred to as the **total instance PGA**, or just **instance PGA**. With Database Control, you set the size of the instance PGA, not individual PGAs.

About Instance Startup and Shutdown

After installation, the Oracle instance is started, and the database is open for access by users with database accounts. At some point, you may want to shut down and restart the instance. This section describes the startup and shutdown processes.

The phrases "starting up and shutting down the Oracle instance" are often used interchangeably with "starting up and shutting down the database."

This section contains the following topics:

- [About Administration Privileges for Startup and Shutdown](#)
- [About Instance Startup](#)
- [About Instance Shutdown](#)

See Also:

- ["Shutting Down and Starting Up the Oracle Instance"](#) on page 5-9
- *Oracle Database Concepts*

About Administration Privileges for Startup and Shutdown

To start or shut down the Oracle instance, you 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 start and shut down the database, but have no privileges to access user objects.

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 initially, only user `SYS` or `SYSTEM` can connect with the `SYSOPER` privilege. Therefore, until you grant the `SYSOPER` privilege to other

users, you must connect to the Oracle instance as user `SYS` or `SYSTEM` to start and shut down the instance. When connecting (logging in) as user `SYS`, you must always specify that you are connecting `AS SYSDBA`.

See Also:

- ["About Administrative Accounts and Privileges"](#) on page 7-4 for more information about the `SYS` and `SYSTEM` users, and the `SYSDBA` privilege

About Instance Startup

When you start the Oracle instance, you typically start it such that the state of the database is `OPEN` and ready for user connections. However, there are situations in which you may want to start the instance with the database in the `MOUNTED` state, but not open. An instance can also be started without the database either mounted or open. Thus, there are three stages to starting an instance:

1. You start the instance using one of the following methods:
 - Using Oracle Enterprise Manager. See ["Shutting Down and Starting Up the Oracle Instance"](#) on page 5-9.
 - Using the SQL*Plus `STARTUP` command. See *Oracle Database Administrator's Guide*.
 - On Microsoft Windows, using the Services program in Control Panel to start the Oracle Database services. See ["Shutting Down and Starting Up Using the Windows Services Program"](#) on page 5-11.

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

2. If you mount the database, then the Oracle instance opens the control file for the database, but does not open the data files. The database is now considered to be in the `MOUNT` state. This state enables you to perform certain administrative functions that cannot be performed when other users are accessing the database. An example of such a function is enabling or disabling the archiving of online redo log files. See ["About Archived Redo Log Files"](#) on page 6-5 for information about online redo log file archiving.
3. If you open the database, then, after reading the parameter file and control file, the online redo log files and data files for the database are also opened. The state of the database is now `OPEN` and user access to the data is available.

The default startup mode for the database (`OPEN`) completes the three stages in sequence. Unless you explicitly specify otherwise, the instance is started, the database is mounted, and then the database is opened.

About Instance Shutdown

Instance shutdown is the reverse of instance startup. When you shut down the Oracle instance, the default mode is a `NORMAL` shutdown, which means users are not allowed to create new connections to the database, but the shutdown process waits for all currently connected users to exit their sessions. After all the users have disconnected, then the committed transactions are written to disk, the database files are closed, and the instance is stopped. However, there are situations in which you may not want to wait for users to disconnect on their own (`IMMEDIATE` mode), or you want to let the current transactions for each user complete before they are disconnected (`TRANSACTIONAL` mode). In emergency situations you can even shut down the

database without waiting for the committed transactions to be written to disk (ABORT mode).

Shutting down an instance goes through the following stages:

1. After all the users have exited from their sessions, or been disconnected, Oracle Database writes data in the SGA to the data files and online redo log files. A checkpoint is performed on the data files and their headers are marked current as of the time of the instance shutdown. The data files and online redo log files are then closed and the state of the database is changed to `CLOSED`. The control file remains open to the instance.
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 dismounted. The instance is in the `NOMOUNT` state.
3. The Oracle instance stops the background processes and deallocates the shared memory used by the SGA.

If a `SHUTDOWN ABORT` or abnormal termination occurs, then the instance of an open database closes and shuts down the database instantaneously. Oracle Database does not write data in the buffers of the SGA to the data files and redo log files. The subsequent reopening of the database requires instance recovery, which Oracle Database performs automatically.

Shutting Down and Starting Up the Oracle Instance

This section provides instructions about two methods you can use to start or shut down the Oracle instance:

- [Shutting Down and Starting Up Using Oracle Enterprise Manager Database Control](#)
- [Shutting Down and Starting Up Using the Windows Services Program](#)

See Also:

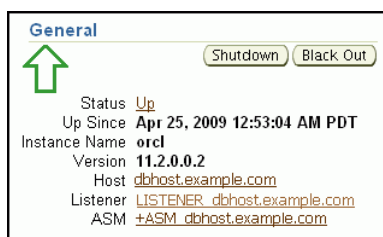
- ["About Instance Startup and Shutdown"](#) on page 5-7

Shutting Down and Starting Up Using Oracle Enterprise Manager Database Control

You can shut down and start the Oracle instance from the Database Home page of Oracle Enterprise Manager Database Control (Database Control).

To shut down and start the Oracle instance using Database Control:

1. Go to the Database Home page.
- See ["Accessing the Database Home Page"](#) on page 3-6.
2. In the General section of the Database Home page, click **Shutdown**.



The Startup/Shutdown Credentials page appears.

Database Instance: orcl > Logged in As SYS

Startup/Shutdown: Specify Host and Target Database Credentials

Specify the following credentials in order to change the status of the database.

Host Credentials

Specify the OS user name and password to login to target database machine.

* Username

* Password

Database Credentials

Specify the credentials for the target database.
To use OS authentication, leave the user name and password fields blank.

* Username

* Password

Database **orcl**

* Connect As

Save as Preferred Credential


Note that you need to login to the database as SYSDBA or SYSOPER in order to change the status of the database.

3. Enter credentials as follows:
 - a. Enter the host computer credentials of the user who installed Oracle Database, or of any user who is authorized to use SQL*Plus.
 - b. Enter the database credentials consisting of the user name `SYS` and the password that you assigned to `SYS` during the installation.
 - c. In the Connect As list, choose the value **SYSOPER**.
4. (Optional) Select the **Save as Preferred Credential** option if you want these credentials to be automatically filled in for you the next time that this page appears.
5. Click **OK**.
A confirmation page appears.
6. Click **Yes** to start the `SHUTDOWN IMMEDIATE` operation on the database.
The Startup/Shutdown: Activity Information page appears, informing you that the database is being shut down.
7. After a short period (approximately 2 minutes), click **Refresh** to be returned to the Database Home page.
The Database Home page indicates that the database instance status is Down.

Database Instance: orcl.example.com

Enterprise Manager is not able to connect to the database instance. The state of the components are listed below. Page Refreshed **Apr 30, 2009 8:55:13 AM PDT**

Database Instance

 Status **Down** Details **There has been a user-initiated shutdown.**

Host **dbhost.example.com**

Port **1521**

SID **orcl**

Oracle Home **/u01/app/oracle/product/11.2.0/dbhome_1**

8. Click **Startup** to start the database.
The Startup/Shutdown Credential page appears again.
9. Enter the credentials, and then click **OK**.
A confirmation page appears.

10. Click Yes.

The Startup page appears, indicating that the database is being started up. When startup is complete, the Login page appears.

11. Log in to the database (and to Database Control).

The Database Home page appears indicating that the database instance status is Up.

Shutting Down and Starting Up Using the Windows Services Program

On Microsoft Windows, you can also start and shut down your Oracle database using the Services program in Control Panel. You must start or stop the following services:

- OracleServiceSID, which is your Oracle instance.
- OracleORACLE_HOMETNSListener, which is your listener. The listener is required for clients to connect to your database.
- OracleDBConsoleSID, which enables clients to connect to Database Control.

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

To start or stop Oracle Database services:

1. Do one of the following:
 - On Windows XP and Windows Server 2003, click **Start**, and then select **Control Panel**.
 - On Windows 2000 Server, click **Start**, select **Settings**, and then select **Control Panel**.

The Control Panel window opens.

2. Double-click the **Administrative Tools** icon, and then double-click the **Services** icon.

The Services window opens, displaying all Windows services that are available on your system.

3. Locate the Oracle Database services listed at the beginning of this section. For example, if your SID is `orcl`, then locate the following services:
 - OracleServiceORCL
 - OracleOraDb11g_home1TNSListener
 - OracleDBConsoleorcl
4. Start or stop all three services, using the following steps for each service:
 - a. Select the service name.
 - b. In the Action menu, click **Start** or **Stop**.

Viewing and Modifying Initialization Parameters

This section provides instructions about viewing the initialization parameter settings for your database and modifying these parameters. You can modify the initialization parameters for the database in one of three ways:

- Until the instance is shut down: The new values for the initialization parameters are applied to the currently running instance, but, when the database is restarted, the initialization parameter values revert to their previous settings.
- From now until the initialization parameter is changed again: The changes are applied to the currently running instance and are also stored in the server parameter file. The changes made to the initialization parameters persist when the database is restarted.
- When the database is restarted: The new values for the initialization parameters are recorded in server parameter file, but are not applied to the currently running instance. The changes take effect only when the database is restarted.

Note: These three scenarios correspond to using the SCOPE=MEMORY, SCOPE=BOTH, and SCOPE=SPFILE clauses of the ALTER SYSTEM SQL statement, respectively, when you use the ALTER SYSTEM statement to change initialization parameters.

To view or modify initialization parameters:

1. At the top of the Database Home page, click **Server** to view the Server subpage.
2. Under Database Configuration, click **Initialization Parameters**.

Oracle Enterprise Manager Database Control (Database Control) displays the Initialization Parameters page.

Database Instance: orcl.example.com > Logged in As SYSTEM
Show SQL Revert Apply

Initialization Parameters

Current SPFile

The parameter values listed here are currently used by the running instance(s).

Name Basic Modified Dynamic Category

Filter on a name or partial name

Apply changes in current running instance(s) mode to SPFile. For static parameters, you must restart the database.

Name ▲	Help	Revisions	Value	Comments	Type
audit_file_dest			/u01/app/oracle/admin/orcl/adu	<input type="text"/>	String
audit_trail			DB		String
diagnostic_dest			/u01/app/oracle	<input type="text"/>	String
dispatchers			(PROTOCOL=TCP) (SERVICE=	<input type="text"/>	String
spfile			+DATA/orcl/spfileorcl.ora	<input type="text"/>	String
compatible			11.2.0.0.0		String

The Initialization Parameters page has two subpages:

- **Current**—This subpage (the default) displays all initialization parameter values that are currently active (in memory) for the Oracle instance.
- **SPFile**—This subpage displays initialization parameter settings in the server parameter file. This subpage is present only when the current instance started up with a server parameter file. The file location is displayed at the top of the subpage.

See the online Help for the Initialization Parameters page for field descriptions.

3. (Optional) On either subpage, reduce the number of initialization parameters displayed by doing one or both of the following, and then clicking **Go**:

- In the **Name** field, enter text.
- Select from one or more of the lists next to the **Name** field.

For example, to view only initialization parameters that have the text `DEST` anywhere in the parameter name, enter `dest` in the **Name** field, and then click **Go**.

4. To modify one or more initialization parameters for the currently running instance only, with the modifications being lost when the instance is restarted, complete the following steps:

- a. On the Current subpage, in the **Value** column, enter new values for the initialization parameters.

Note: If the Value column cannot be written to for a particular initialization parameter, then it means that this parameter is not dynamic—that is, it cannot be changed for the current instance.

- b. Ensure that **Apply changes in current running instance(s) mode to SPFile** is not selected.
- c. (Optional) In the **Comments** column, enter text explaining the reasons for the changes.
- d. Click **Apply**.

A confirmation message appears.

5. To modify initialization parameters for the currently running instance, and also record the modifications in the server parameter file that will persist when the database is restarted, complete the following steps:

- a. On the Current subpage, in the **Value** column, enter new values for the initialization parameters.
- b. Select **Apply changes in current running instance(s) mode to SPFile**.
- c. (Optional) In the **Comments** column, enter text explaining the reasons for the changes.
- d. Click **Apply**.

A confirmation message appears.

6. To modify initialization parameters in the server parameter file only, such that the current instance is not affected and changes take effect only when the database is next restarted, complete the following steps:

- a. Click **SPFile** to view the SPFile subpage.
- b. (Optional) Reduce the number of entries in the initialization parameter list as described in Step 3.
- c. In the **Value** column, enter new values for the initialization parameters.
- d. (Optional) In the **Comments** column, enter text explaining the reasons for the changes.
- e. Click **Apply**.

A confirmation message appears.

Note: Changes to initialization parameters are recorded in the alert log as ALTER SYSTEM statements. See *Oracle Database Administrator's Guide* for information about the alert log.

See Also:

- ["About Initialization Parameters"](#) on page 5-2

Managing Memory

This section provides background information about managing memory for the Oracle instance, and includes instructions about how to adjust the memory allocation for the Oracle instance. It contains the following topics:

- [About Memory Management](#)
- [Enabling Automatic Memory Management](#)
- [Modifying Memory Settings – Automatic Memory Management](#)
- [Modifying Memory Settings – Automatic Shared Memory Management](#)
- [Modifying Memory Settings – Automatic PGA Memory Management](#)

See Also:

- ["About Instance Memory Structure"](#) on page 5-5

About Memory Management

Memory management involves maintaining optimal sizes for the Oracle instance memory structures as demands on the database change. The memory that must be managed is the System Global Area (SGA) memory and the instance Program Global Area (PGA) memory. The instance PGA memory is the collection of memory allocations for all individual PGAs.

Beginning with Oracle Database 11g release 1 (11.1), you can let the database manage the SGA memory and instance PGA memory completely. You designate only the total memory size to be used by the instance, and Oracle Database dynamically exchanges memory between the SGA and the instance PGA as needed to meet processing demands. This capability is referred to as **automatic memory management**. In this memory management mode, the database also dynamically tunes the sizes of the individual SGA components and the sizes of the individual PGAs.

To have more direct control over the sizes of the SGA and instance PGA, use the Memory Advisors page of Database Control to disable automatic memory management and enable automatic shared memory management.

- With **automatic shared memory management**, you set target and maximum sizes for the SGA. Oracle Database then tunes the total size of the SGA to your designated target, and dynamically tunes the sizes of all SGA components.
- When you enable automatic shared memory management, you also implicitly enable automatic PGA memory management. With **automatic PGA memory management**, you set a target size for the instance PGA. The database then tunes the size of the instance PGA to your target, and dynamically tunes the sizes of individual PGAs.

If you want complete control of individual SGA component sizes, then use the Memory Advisors page of Database Control to disable both automatic memory management and automatic shared memory management. This is called **manual shared memory management**. In this mode, you set the sizes of several individual SGA components, thereby determining the overall SGA size. You then manually tune these individual SGA components on an ongoing basis.

Manual shared memory management mode is intended for experienced DBAs only. Note that in this mode, automatic PGA memory management remains enabled.

Note: Although it is possible to disable automatic PGA memory management, it is not recommended, and cannot be done using the Memory Advisors page of Database Control.

Table 5–3 summarizes the various memory management modes that you can set using the Memory Advisors page of Database Control.

Table 5–3 Oracle Database Memory Management Modes

Memory Management Mode	You Set	Oracle Database Automatically Tunes
Automatic memory management	<ul style="list-style-type: none"> ■ Total memory size for this instance ■ (Optional) Maximum memory size for this instance 	<ul style="list-style-type: none"> ■ Total SGA size ■ SGA component sizes ■ Instance PGA size ■ Individual PGA sizes
Automatic shared memory management and automatic PGA memory management (Automatic memory management disabled)	<ul style="list-style-type: none"> ■ SGA target size ■ (Optional) SGA maximum size ■ Instance PGA target size 	<ul style="list-style-type: none"> ■ SGA component sizes ■ Individual PGA sizes
Manual shared memory management and automatic PGA memory management (Automatic memory management and automatic shared memory management disabled)	<ul style="list-style-type: none"> ■ Shared pool size ■ Buffer cache size ■ Java pool size ■ Large pool size ■ Instance PGA target size 	<ul style="list-style-type: none"> ■ Individual PGA sizes

Note: Automatic Memory Management is not available on all platforms. See *Oracle Database Administrator's Guide* for more information about supported platforms.

If you choose the basic installation option when you install the database, then automatic memory management is enabled. If you choose advanced installation, then DBCA enables you to select from the three memory management modes. Oracle recommends that you enable automatic memory management.

Whichever memory management mode you choose, you may have occasion to adjust memory settings as demands on the database or on its host computer change. Reasons why you adjust memory settings include the following:

- You receive a memory-related alert or error message.

- You receive a memory-related recommendation from Automatic Database Diagnostic Monitor (ADDM).
- You want to change the amount of memory allocated to accommodate future growth in memory demand.

You can use a *memory advisor* to help you adjust memory sizes. See "[Modifying Memory Settings – Automatic Shared Memory Management](#)" on page 5-22 for an example of using a memory advisor.

See Also:

- "[Enabling Automatic Memory Management](#)" on page 5-18
- "[Modifying Memory Settings – Automatic Memory Management](#)" on page 5-19
- "[Modifying Memory Settings – Automatic PGA Memory Management](#)" on page 5-25

Enabling Automatic Memory Management

If you did not enable automatic memory management when you installed and configured your database, then Oracle recommends that you do so after installation, unless you are an experienced DBA with specific reasons to manually tune memory sizes. With automatic memory management, the Oracle instance dynamically tunes all memory components to optimize performance as the workload changes.

To enable automatic memory management:

1. Go to the Database Home page, logging in as user `SYS`.

See "[Accessing the Database Home Page](#)" on page 3-6.

Note: You must log in as user `SYS` to change the maximum memory size, which is a static initialization parameter.

2. At the top of the page, click **Server**.
The Server subpage appears.
3. In the Database Configuration section, click **Memory Advisors**.
The Memory Advisors page appears with the SGA subtab (in the middle of the page) displayed by default.
4. If the Maximum SGA Size field contains a positive number that is greater than or equal to the desired total amount of memory to allocate to the database, then skip to Step 10.
5. In the **Maximum SGA Size (MB)** field, enter the maximum permissible size for database memory.
Use the sum of the current sizes of the SGA and instance PGA as a guideline, and optionally add some room for growth.
6. Click **Apply**.
A confirmation page appears, indicating that you must restart the database.
7. Click **Yes** and follow the guided instructions to restart the database.

8. After you have initiated the database restart, wait a few minutes and then click **Refresh**.
The Database Home page appears.
9. Return to the Memory Advisors page.
You might need to log out and log back in to Enterprise Manager to get this page to display correctly after the database restart.
10. Next to Automatic Memory Management **Disabled**, click **Enable**.
The Enable Automatic Memory Management page appears.
11. In the field titled **Total Memory Size for Automatic Memory Management**, enter the desired amount of memory to allocate to the database, and then click **OK**.
A confirmation message appears.

See Also:

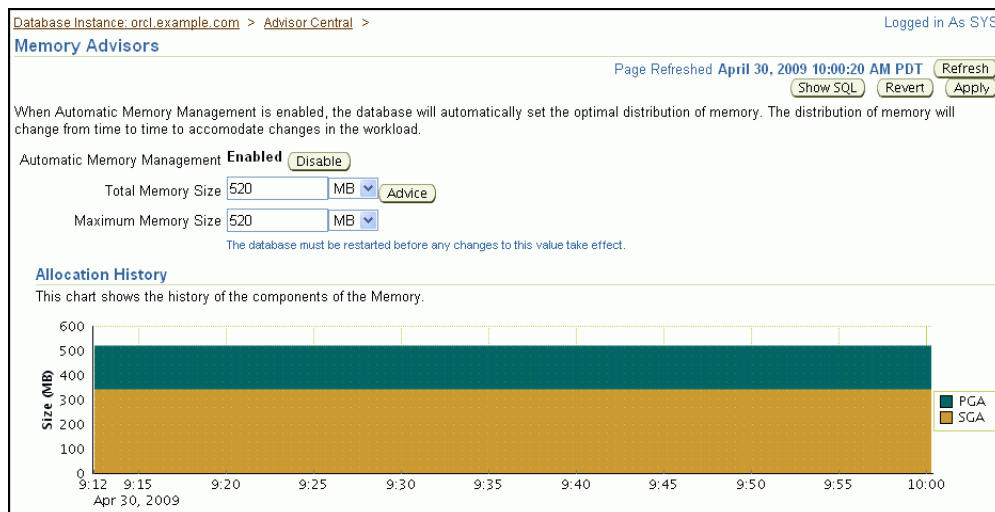
- ["About Memory Management"](#) on page 5-15

Modifying Memory Settings – Automatic Memory Management

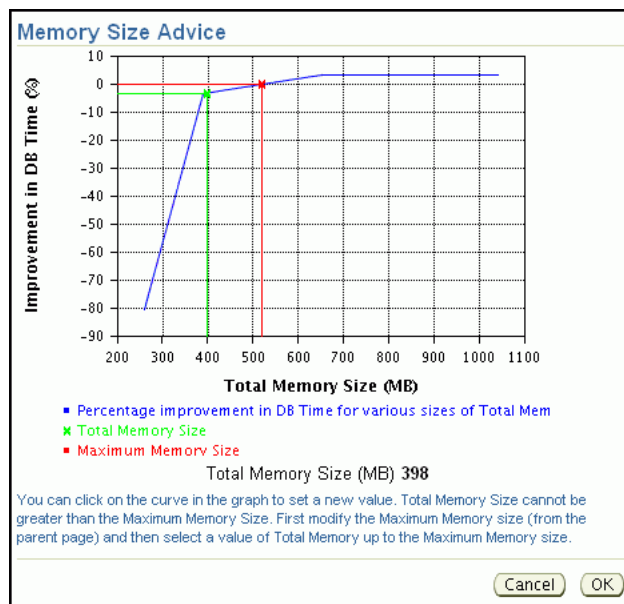
Modifying memory settings for automatic memory management involves using the Memory Advisors page of Database Control to change the total memory size setting. The procedure for changing this setting differs depending on whether or not your desired new setting exceeds the current setting for maximum memory size. In either case, the steps involved require that you must first have enabled Automatic Memory Management. See ["Enabling Automatic Memory Management"](#) on page 5-18.

To modify total memory size (new value does not exceed maximum memory size):

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Server**.
3. In the Database Configuration section, click **Memory Advisors**.
The Memory Advisors page appears.



4. Do one of the following:
 - In the **Total Memory Size** field, enter a new value that is less than or equal to the value in the current Maximum Memory Size field.
 - Click **Advice** to use the Memory Advisor to help you choose a new size.
The Memory Size Advice page appears.



5. If you chose to use the Memory Advisor, then click a point on the curve to change the total memory size. Otherwise, go to Step 7.
The new total memory size is displayed below the graph.
6. Continue clicking the curve until the desired value is displayed, then click **OK**.
You are returned to the Memory Advisors page.
7. Click **Apply**.
A confirmation message appears.

To modify total memory size (new value exceeds maximum memory size):

1. Go to the Database Home page, and log in as user `SYS` with the `SYSDBA` privilege.
See "[Accessing the Database Home Page](#)" on page 3-6.

Note: You must log in as user `SYS` to change the maximum memory size, which is a static initialization parameter.

2. At the top of the page, click **Server**.
3. In the Database Configuration section, click **Memory Advisors**.
The Memory Advisors page appears.
4. In the **Maximum Memory Size (MB)** field, enter a new value for the maximum total memory to allocate to the database.

This value must be greater than or equal to the desired new value for total memory size.

5. Click **Apply**.

A confirmation page appears, indicating that you must restart the database.

6. Click **Yes** and follow the guided instructions to restart the database.

7. After you have initiated the database restart, wait a few minutes, then click **Refresh**.

The Database Home page appears.

8. Return to the Memory Advisors page.

9. Do one of the following:

- In the **Total Memory Size** field, enter a new value that is less than or equal to the value in the current Maximum Memory Size field.
- Click **Advice** to use the Memory Advisor to help you choose a new size.

The Memory Size Advice page appears.

10. If you chose to use the Memory Advisor, then click a point on the curve to change the total memory size. Otherwise, go to Step 12.

The new size is displayed below the graph.

11. Continue clicking the curve until the desired value is displayed, then click **OK**.

You are returned to the Memory Advisors page.

12. Click **Apply**.

A confirmation message appears.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about memory parameters and their tuning
- ["Optimizing Memory Usage with the Memory Advisors"](#) on page 10-38
- ["About Memory Management"](#) on page 5-15

Modifying Memory Settings – Automatic Shared Memory Management

Modifying memory settings for automatic shared memory management involves using the Memory Advisors page of Database Control to change the total SGA size. This section assumes that automatic memory management is disabled, that automatic shared memory management is enabled, and that the desired new value for the total SGA size does not exceed the current value for maximum SGA size.

Note: If you want a new total SGA size that exceeds the current value for maximum SGA size, then you must first log in as user SYS, change the maximum SGA size, and restart the database. See ["To modify total memory size \(new value exceeds maximum memory size\):"](#) on page 5-21 for a similar procedure.

To modify the total SGA memory size:

1. Go to the Database Home page.

See "Accessing the Database Home Page" on page 3-6.

2. At the top of the page, click **Server** to view the Server subpage.
3. In the Database Configuration section, click **Memory Advisors**.

The Memory Advisors page appears. Scroll down to the SGA subpage.

SGA PGA

The System Global Area (SGA) is a group of shared memory structures that contains data and control information for one Oracle database. The SGA is allocated in memory when an Oracle database instance is started.

Allocation History
This chart shows the history of the components of the SGA.

Current Allocation

Automatic Shared Memory Management **Enabled** Disable
 Total SGA Size (MB) Advice

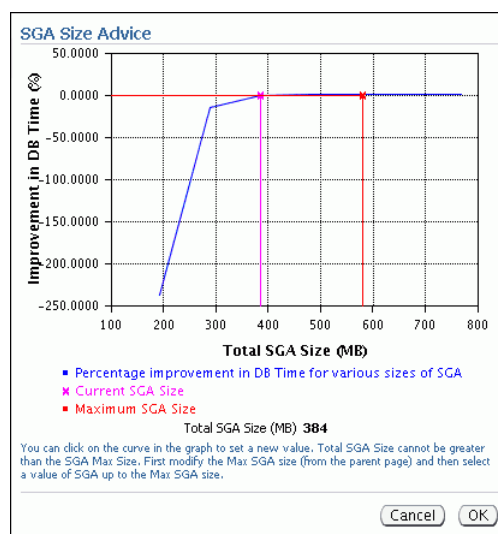
SGA Component Current Allocation (MB)	
Shared Pool	192
Buffer Cache	112
Large Pool	4
Java Pool	4
Other	16

Maximum SGA Size
The Maximum SGA Size specifies the maximum memory that the database may allocate. If you specify the Maximum SGA Size, you can later dynamically change the Total SGA Size above (provided Total SGA Size does not exceed the Maximum SGA Size).

Maximum SGA Size (MB)

4. In the Current Allocation section, do one of the following:
 - In the **Total SGA Size (MB)** field, enter a new total size for the SGA.
 - Click **Advice** to use the SGA Advisor to help you choose a new SGA size.

The SGA Size Advice window opens.



Click a point on the curve to change the total SGA size. The new SGA size is displayed below the graph. Continue clicking the curve until the desired value is displayed, and then click **OK**.

This example graph shows that at the current time, increasing the total SGA size is not expected to provide any gain in performance.

Note: The settings on this page can be modified only if Automatic Memory Management is disabled.

5. If you chose to use the SGA Size Advisor, then click a point on the curve to change the total memory size. Otherwise, go to Step 7.

The new size is displayed below the graph.

6. Continue clicking the curve until the desired value is displayed, then click **OK**.

You are returned to the Memory Advisors page.

7. Click **Apply**.

A confirmation message appears.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about memory parameters and their tuning
- ["Optimizing Memory Usage with the Memory Advisors"](#) on page 10-38
- ["About Memory Management"](#) on page 5-15

Modifying Memory Settings – Automatic PGA Memory Management

Modifying memory settings for automatic PGA memory management involves using the Memory Advisors page of Database Control to modify the instance PGA size. This section assumes that automatic memory management is disabled, and that automatic PGA memory management is enabled.

Note: When you disable automatic memory management, automatic PGA memory management is enabled by default.

To modify the instance PGA size:

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Database Configuration section, click **Memory Advisors**.
The Memory Advisors page appears, with the SGA subpage shown.
4. Click **PGA** to view the PGA subpage.

SGA	PGA
<p>The Program Global Area (PGA) is a memory buffer that contains data and control information for a server process. A PGA is created by Oracle when a server process is started.</p>	
Aggregate PGA Target: <input type="text" value="188"/> MB <input type="button" value="Advice"/>	
Current PGA Allocated (KB): 99285	
Maximum PGA Allocated (KB): 136450 <small>(since startup)</small>	
Cache Hit Percentage (%): 100	
<input type="button" value="PGA Memory Usage Details"/>	
<input checked="" type="checkbox"/> TIP The sum of PGA and SGA should be less than the total system memory minus memory required by the operating system and other applications.	

5. Do one of the following:

- In the **Aggregate PGA Target** field, enter a new size for the instance PGA.
- Click **Advice** to use the PGA Advisor to help you choose a new instance PGA size.

See *Oracle Database Performance Tuning Guide* for information about sizing the instance PGA.

6. When you have specified the new instance PGA size, click **Apply**.

A confirmation message appears.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about memory parameters and their tuning
- "[About Memory Management](#)" on page 5-15

Instances: Oracle By Example Series

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

To view the Instances OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/instance/instance.htm

Managing Database Storage Structures

This chapter discusses using Oracle Enterprise Manager Database Control (Database Control) to view and manage the storage structures of your database. This chapter contains the following sections:

- [About Database Storage Structures](#)
- [Viewing Database Storage Structure Information](#)
- [Performing Common Database Storage Tasks](#)
- [Managing the Online Redo Log](#)
- [Managing Undo Data](#)
- [Storage: Oracle By Example Series](#)

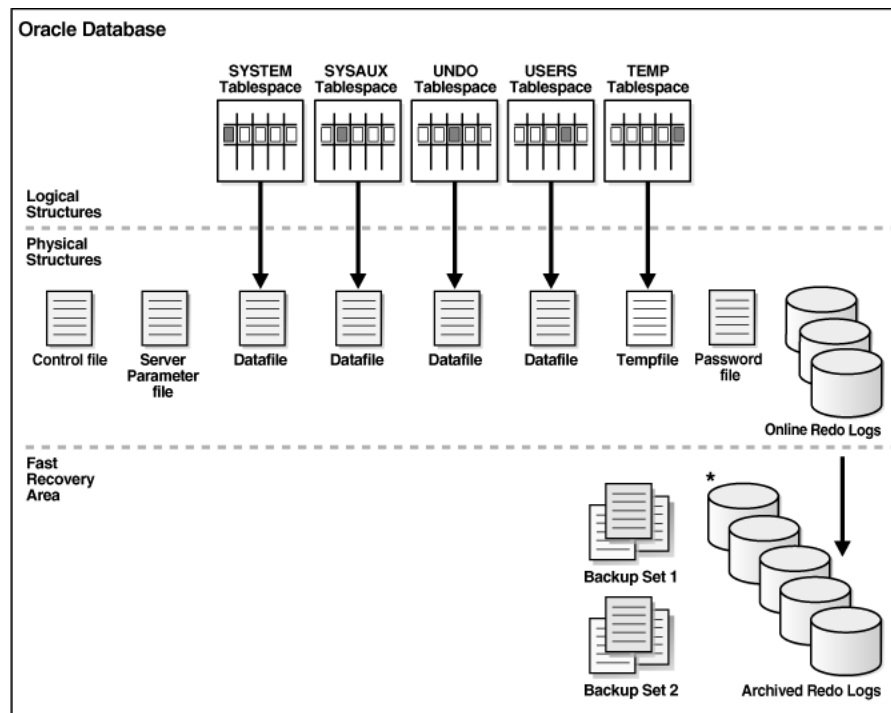
About Database Storage Structures

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

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

[Figure 6-1](#) on page 6-2 shows the relationships between logical and physical structures. This figure also shows recovery-related structures that are optionally kept in the fast recovery area. See "[Fast Recovery Area](#)" on page 9-4 for more information.

Figure 6–1 Oracle Database Storage Structures



* Archived Redo Logs present only after turning on log archiving (ARCHIVELOG mode)

Oracle Database can automate much of the management of its structure. Oracle Enterprise Manager Database Control (Database Control) provides a Web-based graphical user interface (GUI) to enable easier management and monitoring of your database.

To view a database storage structure, go to the Storage section of the Server subpage and click on any of the links to access the storage pages.



This section provides background information about the various database storage structures. It contains the following topics:

- [About Control Files](#)
- [About Online Redo Log Files](#)
- [About Archived Redo Log Files](#)
- [About Rollback Segments](#)
- [About Data Files](#)

- [About Tablespaces](#)
- [About Other Storage Structures](#)

See Also:

- ["Viewing Database Storage Structure Information"](#) on page 6-11
- *Oracle Database Concepts*

About Control Files

A control file tracks the physical components of the database. It is the root file that the database uses to find all the other files used by the database. Because of the importance of the control file, Oracle recommends that the control file be **multiplexed**, or have multiple identical copies. For databases created with Oracle Database Configuration Assistant (DBCA), three copies of the control file are automatically created and kept synchronized with each other.

If any control file fails, then your database becomes unavailable. If 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 using the remaining control files.

See Also:

- *Oracle Database Administrator's Guide* for detailed information about control files
- ["Viewing Control File Information"](#) on page 6-11
- *Oracle Database Concepts*

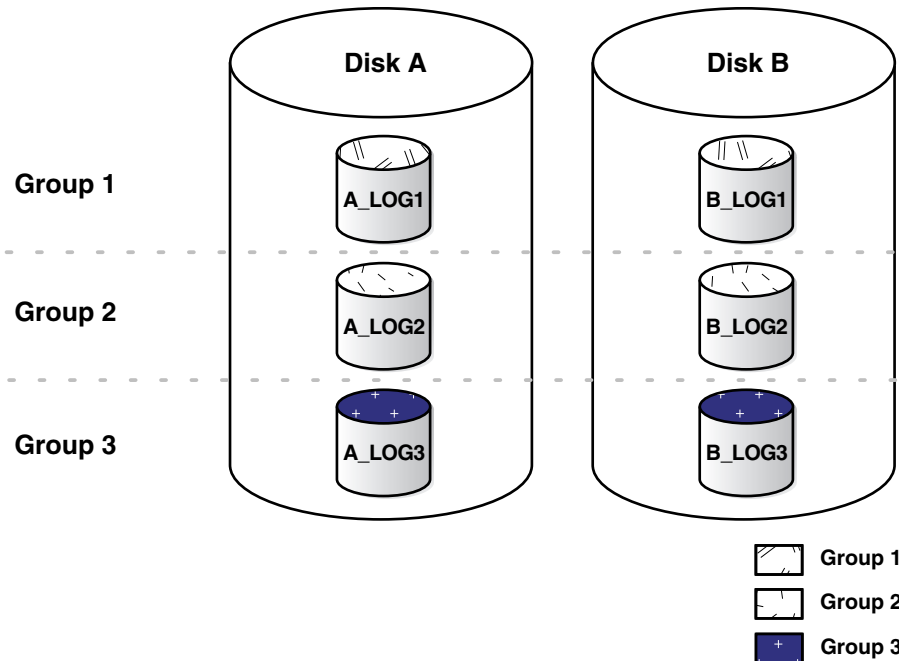
About Online Redo Log Files

Every Oracle database has a set of two or more online redo log files. The set of online 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 online redo log stores a copy of the changes made to data. If a failure requires a data file to be restored from backup, then the recent data changes that are missing from the restored data file can be obtained from the online redo log files, so work is never lost. The online redo log files are used to recover a database after hardware, software, or media failure. To protect against a failure involving the online redo log file itself, Oracle Database can multiplex the online redo log file so that two or more identical copies of the online redo log file can be maintained on different disks.

The online redo log for a database consists of groups of online redo log files. A group consists of an online redo log file and its multiplexed copies. Each identical copy is considered to be a member of that group. Each group is defined by a number, such as Group 1.

[Figure 6-2](#) on page 6-4 shows the configuration of a database that has three online redo log groups and two members in each group. For each group, the members are stored on separate disks for maximum availability. For example, the members of Group 1 are the redo log files `A_LOG1` and `B_LOG1`.

Figure 6–2 Online Redo Log Groups and Their Members

The database log writer process (LGWR) writes redo records from the memory buffer to a redo log group until the log files in that group reach their storage size limit, or until you request a log switch operation. The LGWR process then writes to the next log group. The LGWR process 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
- ["Viewing Online Redo Log File Information"](#) on page 6-12
- ["Switching a Log File"](#) on page 6-29
- ["Managing the Online Redo Log"](#) on page 6-28
- *Oracle Database Concepts*

About Archived Redo Log Files

When you enable archiving of the online redo logs, Oracle Database copies the online redo log files to another location before they are overwritten. These copied files are referred to as **archived redo log files**. You can archive to multiple locations.

These archived redo log files extend the amount of redo data that can be saved and are used for recovery. Archived redo log files are required to recover a backup of the database from the time of the backup to the current time. Archiving can be either enabled or disabled for the database, but Oracle strongly recommends that you enable archiving. Oracle also recommends that you configure the database to write archived redo log files to the fast recovery area.

See Also:

- *Oracle Database Administrator's Guide* for detailed information about archived redo log files
- ["Viewing Archived Redo Log File Information"](#) on page 6-12
- ["Configuring Recovery Settings"](#) on page 9-7 for information about enabling redo log archiving
- ["Fast Recovery Area"](#) on page 9-4 for background information about the fast recovery area
- *Oracle Database Concepts*

About Rollback Segments

Rollback segments were database structures used to track undo information for the database in earlier releases of Oracle Database. Now, the preferred way of managing undo information is with the undo tablespace. For more information, see ["Managing Undo Data"](#) on page 6-30.

Note: Oracle Database uses a `SYSTEM` rollback segment for performing system transactions. It is created automatically when the database is created, and is always brought online at instance startup. It is located in the `SYSTEM` tablespace. You are not required to perform any operations to manage the `SYSTEM` rollback segment.

About Data Files

Data files are the operating system files that store 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 data files that are associated only with temporary tablespaces.

Data files can be broken down into the following components:

- Segment
A **segment** contains a specific type of database object. For example, a table is stored in a table segment, and an index is stored in an index segment. A data file can contain many segments.
- Extent
An **extent** is a contiguous set of data blocks within a segment. Oracle Database allocates space for segments in units of one extent. When the existing extents of a segment are full, the database allocates another extent for that segment.
- Data block
A **data block**, also called a **database block**, is 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. It is possible, however, to create a tablespace with a block size different than the default block size.

Segments, extents, and data blocks are all logical structures. Only Oracle Database can determine how many data blocks are in a file. The operating system recognizes only

files and operating system blocks, not the number of data blocks in an Oracle Database file. Each data block maps to one or more operating system blocks.

See Also:

- *Oracle Database Administrator's Guide* for detailed information about data files
- ["Viewing Data File Information"](#) on page 6-13
- *Oracle Database Concepts* for more information about segments, extents, and blocks

About Tablespaces

A database is divided into logical storage units called **tablespaces**, which group related logical structures (such as tables, views, and other database objects). For example, all application objects can be grouped into a single tablespace to simplify maintenance operations.

A tablespace consists of one or more physical data files. Database objects assigned to a tablespace are stored in the physical data files of that tablespace.

When you create an Oracle database, some tablespaces already exist, such as `SYSTEM` and `SYSAUX`.

Tablespaces provide a means to physically locate data on storage. When you define the data files that comprise a tablespace, you specify a storage location for these files. For example, you might specify a data file location for a certain tablespace as a designated host directory (implying a certain disk volume) or designated Oracle Automatic Storage Management disk group. Any schema objects assigned to that tablespace then get located in the specified storage location. Tablespaces also provide a unit of backup and recovery. The backup and recovery features of Oracle Database enable you to back up or recover at the tablespace level.

[Table 6–1](#) describes some tablespaces included in the database.

Table 6–1 Tablespaces and Descriptions

Tablespace	Description
EXAMPLE	This tablespace contains the sample schemas that are included with Oracle Database. The sample schemas provide a common platform for examples. Oracle documentation and educational materials contain examples based on the sample schemas.
SYSTEM	This tablespace is automatically created at database creation. Oracle Database 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 <code>SYS</code> schema, and can be accessed only by the <code>SYS</code> user or other administrative users with the required privilege.
SYSAUX	This is an auxiliary tablespace to the <code>SYSTEM</code> tablespace. The <code>SYSAUX</code> tablespace contains data for some components and products, reducing the load on the <code>SYSTEM</code> tablespace. Every database using Oracle Database 10g release 1 (10.1) or later must have a <code>SYSAUX</code> tablespace. Components that use <code>SYSAUX</code> as their default tablespace during installation include Automatic Workload Repository, Oracle Streams, Oracle Text, and Database Control Repository. For more information, see <i>Oracle Database Administrator's Guide</i> .

Table 6–1 (Cont.) Tablespaces and Descriptions

Tablespace	Description
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. If no temporary tablespace is specified when a user account is created, then Oracle Database assigns this tablespace to the user.
UNDOTBS1	This is the undo tablespace used by the database to store undo information. See "Managing Undo Data" on page 6-30 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. Similar to 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 designated as the default tablespace for all new users.

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

- [Locally Managed Tablespaces](#)
- [Tablespace Types](#)
- [Tablespace Status](#)
- [Autoextend Tablespace](#)
- [Encrypted Tablespaces](#)

See Also:

- ["Viewing Tablespace Information"](#) on page 6-14
- ["Creating a Tablespace"](#) on page 6-14
- ["Modifying a Tablespace"](#) on page 6-18
- *Oracle Database Concepts*

Locally Managed Tablespaces

Space management within a tablespace involves keeping track of available (free) and used space, so that space is allocated efficiently during data insertion and deletion. Locally managed tablespaces keep the space allocation information within the tablespace, not in the data dictionary, thus offering better performance. By default, Oracle Database sets all newly created tablespaces to be locally managed with automatic segment management, a feature that further improves performance.

Tablespace Types

There are three types of tablespaces:

- **Permanent**
You use permanent tablespaces to store your user and application data. Oracle Database uses permanent tablespaces to store permanent data, such as system data. Each user is assigned a default permanent tablespace.
- **Undo**

A database running in automatic undo management mode transparently creates and manages undo data in the undo tablespace. Oracle Database uses undo data to roll back transactions, to provide read consistency, to help with database recovery, and to enable features such as Oracle Flashback Query. A database instance can have only one active undo tablespace.

- **Temporary**

Temporary tablespaces are used for storing temporary data, as would be created when SQL statements perform sort operations. An Oracle database gets a temporary tablespace when the database is created. You would create another temporary tablespace if you were creating a temporary tablespace group. Under typical circumstances, you do not have to create additional temporary tablespaces. If you have an extremely large database, then you might configure additional temporary tablespaces.

The physical files that comprise a temporary tablespace are called tempfiles, as opposed to data files.

The `TEMP` tablespace is typically used as the default temporary tablespace for users who are not explicitly assigned a temporary tablespace.

See Also:

- *Oracle Database Administrator's Guide* to learn more about temporary tablespaces
- *Oracle Database Concepts* for more information about undo tablespaces

Tablespace Status

You can set tablespace status 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 would 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 has a status of Offline, then no users can access it. You might change the status of a tablespace to Offline before performing maintenance or recovery on the data files associated with that tablespace.

Autoextend Tablespace

You can set a tablespace to automatically extend itself by a specified amount when it reaches its size limit. If you do not enable autoextend, then you are alerted when the tablespace reaches its critical or warning threshold size. The critical and warning threshold parameters have default values that you can change at any time. These parameters also cause alerts to be generated for autoextending tablespaces that are approaching their specified size limit. You can respond to size alerts by manually increasing the tablespace size. You do so by increasing the size of one or more of the tablespace data files or by adding another data file to the tablespace.

Note: Although it is common to refer to tablespaces as autoextending, automatic extension is a data file property, not a tablespace property. That is, when you create the data files that comprise a tablespace, you indicate whether these data files automatically extend. A tablespace that has autoextending data files is considered to be an autoextending tablespace. You can specify a maximum size for an autoextending data file.

Encrypted Tablespaces

Encrypted tablespaces primarily protect your data from unauthorized access by means other than through the database. For example, when encrypted tablespaces are written to backup media for travel from one Oracle database to another or for travel to an off-site facility for storage, they remain encrypted. Also, encrypted tablespaces protect data from users who try to circumvent the security features of the database and access database files directly through the operating system file system.

You can encrypt any permanent tablespace to protect sensitive data. When you encrypt a tablespace, all tablespace blocks are encrypted. All segment types are supported for encryption, including tables, clusters, indexes, LOBs, table and index partitions, and so on. Tablespace encryption is completely transparent to your applications, so no application modification is necessary.

See Also:

- *Oracle Database 2 Day + Security Guide* for more information on securing data by using encryption

About 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 Oracle instance at startup to determine the run-time properties and resources for the database. Some parameters can be set or modified while the database is running. Other initialization parameters require the database to be restarted for the changes to take effect. See "[Viewing and Modifying Initialization Parameters](#)".

Password File

A database can use a password file to authenticate administrative users with SYSDBA or SYSOPER connection privileges. A password file is required for remote connections to the database as SYSDBA or SYSOPER. These privileges enable a DBA to start and shut down the database and perform other high-level administrative tasks. This password file is outside of the database itself, thereby enabling the authentication of a DBA when the database is not yet started. (A DBA must authenticate before starting the database.)

When you invoke DBCA as part of the Oracle Database installation process, DBCA creates a password file with one entry: the SYS user. Granting SYSDBA to a user adds that user to the password file automatically.

Note: Oracle Database can also use operating system authentication to authenticate users with the `SYSDBA` or `SYSOPER` privileges. See ["Starting SQL*Plus and Connecting to the Database"](#).

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.

See Also:

- [Chapter 5, "Managing the Oracle Instance"](#) for more information about initialization parameters and the initialization parameter file
- *Oracle Database Concepts* for more information about password files and operating system authentication
- ["SYSDBA and SYSOPER System Privileges"](#) on page 7-5
- [Chapter 9, "Performing Backup and Recovery"](#) for more information about backup files
- *Oracle Database Concepts*

Viewing Database Storage Structure Information

To assist you in managing the storage structures within your database, this section provides instructions for viewing information about the various database storage structures using Oracle Enterprise Manager Database Control (Database Control).

This section contains the following topics:

- [Viewing Control File Information](#)
- [Viewing Online Redo Log File Information](#)
- [Viewing Archived Redo Log File Information](#)
- [Viewing Data File Information](#)
- [Viewing Tablespace Information](#)

See Also:

- ["About Database Storage Structures"](#) on page 6-1

Viewing Control File Information

You can use Database Control to view location and status information about control files.

To view control file information:

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Storage section, click **Control Files**.

The Control Files page appears, showing the General subpage.

This page shows whether your database has a multiplexed control file. The Advanced and Record Section subpages give you more detailed information about your control files.

See Also:

- ["About Control Files"](#) on page 6-3

Viewing Online Redo Log File Information

You use Database Control to view status and multiplexing information about online redo log files.

To view online redo log file information:

1. Go to the Database Home page.

See ["Accessing the Database Home Page"](#) on page 3-6.

2. At the top of the page, click **Server** to view the Server subpage.

3. In the Storage section, click **Redo Log Groups**.

The Redo Log Groups page appears. This page shows the attributes of the online redo log groups for your database.

When an online redo log group contains only one member, it is not multiplexed. Note the Status attribute for the online redo log groups. The online redo log group with status Current is the one currently being written to disk by the log writer.

4. (Optional) In the **Group** column, click a group number to view information about online redo log group members.

See Also:

- ["About Online Redo Log Files"](#) on page 6-4
- ["Managing the Online Redo Log"](#) on page 6-28

Viewing Archived Redo Log File Information

You use Database Control to view status information about archived redo log files.

Note: Archived redo log files do not exist until you set the database in ARCHIVELOG mode.

To view archived redo log file information:

1. Go to the Database Home page.

See ["Accessing the Database Home Page"](#) on page 3-6.

2. At the top of the page, click **Server** to view the Server subpage.

3. In the Storage section, click **Archive Logs**.

The Archive Logs page appears.

See Also:

- ["About Archived Redo Log Files"](#) on page 6-5
- ["Configuring Your Database for Basic Backup and Recovery"](#) on page 9-5

Viewing Data File Information

You use Database Control to view status and location information about data files.

To view data file information:

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Storage section, click **Datafiles**.
The Datafiles page appears.
4. (Optional) Select a data file, and then click **View** to display more attributes for the data file.
5. (Optional) Click the name of a tablespace in the **Tablespace** column to display the attributes of the tablespace associated with the data file.

See Also:

- ["About Data Files"](#) on page 6-6
-

Viewing Tablespace Information

You use Database Control to view configuration, size, and status information about tablespaces.

To view tablespace information:

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Storage section, click **Tablespaces**.
The Tablespaces page appears.

See Also:

- ["About Tablespaces"](#) on page 6-7
- [Modifying a Tablespace](#)

Performing Common Database Storage Tasks

As data is added to your database, the tablespace requirements for your database change. As a database administrator (DBA), you can use Oracle Enterprise Manager Database Control (Database Control) to perform the following tasks to effectively manage the tablespaces and database storage:

- [Creating a Tablespace](#)
- [Modifying a Tablespace](#)
- [Dropping a Tablespace](#)
- [Reclaiming Unused Space](#)

Creating a Tablespace

You can create additional tablespaces to store user data, so that not all data is stored in the `USERS` tablespace. The following are some reasons to create additional tablespaces:

- For certain users, groups of users, or applications, it may be convenient to keep all application data in a separate tablespace or set of tablespaces for backup and recovery or maintenance reasons. For example, suppose you must recover all application data from backup due to a hardware or software failure, and you want to perform an offline recovery. If the application data is kept in a separate tablespace, then you can take just that tablespace offline and recover it, without affecting the operation of other database applications.
- Some applications, such as those with large partitioned tables, may benefit from distributing data across multiple tablespaces. This approach allows the optimal use of the available storage because frequently accessed data can be placed on high performance disks, and infrequently retrieved data can be placed on less expensive storage.

To create a tablespace:

1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Storage section, click **Tablespaces**.
The Tablespaces page appears.

Tablespaces Object Type:

Search
 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:

Actions:

Select	Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
<input checked="" type="radio"/>	EXAMPLE	332.6	312.8	<div style="width: 94.0%;"></div> 94.0	YES	19.9	<input checked="" type="checkbox"/>	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSAUX	366.9	349.8	<div style="width: 95.3%;"></div> 95.3	YES	17.1	<input checked="" type="checkbox"/>	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	SYSTEM	700.0	563.9	<div style="width: 80.6%;"></div> 80.6	YES	136.1	<input checked="" type="checkbox"/>	1	PERMANENT	LOCAL	MANUAL
<input type="radio"/>	TEMP	40.0	1.0	<div style="width: 2.5%;"></div> 2.5	YES	39.0	<input checked="" type="checkbox"/>	1	TEMPORARY	LOCAL	MANUAL
<input type="radio"/>	TOOLS	100.0	1.0	<div style="width: 1.0%;"></div> 1.0	YES	99.0	<input checked="" type="checkbox"/>	1	PERMANENT	LOCAL	AUTO
<input type="radio"/>	UNDOTBS	290.0	23.3	<div style="width: 8.0%;"></div> 8.0	YES	266.7	<input checked="" type="checkbox"/>	1	UNDO	LOCAL	MANUAL
<input type="radio"/>	USERS	100.0	1.0	<div style="width: 1.0%;"></div> 1.0	YES	99.0	<input checked="" type="checkbox"/>	1	PERMANENT	LOCAL	AUTO

Total Allocated Size (GB) **1.88** Online Offline Read Only
 Total Used (GB) **1.22**
 Total Allocated Free Space (GB) **0.66**

4. Do one of the following:
 - a. To create a new tablespace, click **Create**.
 - b. To create a tablespace that is similar to an existing tablespace, in the **Select** column, select the tablespace whose attributes you want to reproduce. In the Actions list, select **Create Like** and then click **Go**.

The Create Tablespace page appears, showing the General subpage.

Database Instance: orcl.example.com > Tablespaces > Logged in As SYSTEM

Create Tablespace

General Storage

* Name

Extent Management **Type** **Status**

Locally Managed Permanent Read Write

Dictionary Managed Set as default permanent tablespace Read Only

Encryption Offline

Temporary Set as default temporary tablespace

Undo Undo Retention Guarantee: Yes No

Datafiles

Use bigfile tablespace
 Tablespace can have only one datafile with no practical size limit.

Select Name	Directory	Size (MB)
No items found		

5. In the **Name** field, enter a name for the tablespace.
6. In the Extent Management section, select **Locally Managed**.
 See "[Locally Managed Tablespaces](#)" on page 6-8.
7. In the Type section, select **Permanent**.
 See "[Tablespace Types](#)" on page 6-8.

8. In the Status section, select **Read Write**.
See "[Tablespace Status](#)" on page 6-9.
9. In the Datafiles section, click **Add** to add one or more data files to the tablespace.
The Add Datafile page appears.

Note: If you select **Use bigfile tablespace**, then the tablespace can have only one data file. 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.

10. If the Storage Type is File System, then in the **File Name** field, enter a file name. In the **File Directory** and **File Size** fields, enter appropriate values for your data file location and initial size.
11. In the Storage section, complete the following steps:
 - a. Select **Automatically extend data file when full (AUTOEXTEND)**.
 - b. Set a suitable increment, such as 10 MB.
This is the amount of disk space that is added to the data file when it needs more storage space.
 - c. For Maximum File Size, do one of the following, depending on available storage:
 - Select **Unlimited** to permit the file to increase without limits.
 - Select **Value**, and then enter a value in KB, MB, GB, or TB.
12. Click **Continue**.
The Create Tablespace page returns.
13. (Optional) Toward the top of the page, click **Storage** to view the Storage subpage. Examine all of the default settings on this subpage.
14. Click **OK** to add the tablespace.
The Tablespaces page returns, showing the newly created tablespace.

See Also:

- "[About Tablespaces](#)" on page 6-7

Modifying a Tablespace

You can use Database Control to modify a tablespace. For example, you can extend it by increasing data file sizes or adding another data file, set it to automatically extend, change its space usage alert thresholds, or change its status to Offline. When you get a critical or warning alert, you might have to extend a tablespace if the alert is related to a space limitation or you might have to take the tablespace offline for recovery if the alert is related to corrupted data or other serious errors.

This section contains the following topics:

- [Setting a Tablespace to Automatically Extend](#)
- [Changing Space Usage Alert Thresholds for a Tablespace](#)

- [Taking a Tablespace Offline](#)

See Also:

- ["About Tablespaces"](#) on page 6-7

Setting a Tablespace to Automatically Extend

This section explains how to set a tablespace to automatically extend when it reaches its size limit. The following instructions assume that the tablespace was previously not an autoextending tablespace.

To set a tablespace to automatically extend:

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Storage section, click **Tablespaces**.
The Tablespaces page appears.
4. Select the tablespace for which you want to enable autoextend, and then click **Edit**.
The Edit Tablespace page appears.
5. Select the first data file associated with the tablespace, and then click **Edit**.
The Edit Datafile page appears.
6. In the Storage section, complete the following steps:
 - a. Select **Automatically extend data file when full (AUTOEXTEND)**.
 - b. Set a suitable increment, such as 10 MB.
This is the amount of disk space that is added to the data file when it needs more storage space.
 - c. For Maximum File Size, do one of the following, depending on available storage:
 - Select **Unlimited** to permit the file to increase without limits.
 - Select **Value**, and then enter a value in KB, MB, GB, or TB.
7. Click **Continue**.
The Edit Tablespace page returns.
8. Repeat Step 5 through Step 7 for each additional data file associated with the tablespace.
9. Click **Apply**.
A confirmation message appears.

Changing Space Usage Alert Thresholds for a Tablespace

You receive an alert in Database Control when a space usage threshold for a tablespace is reached. There are two types of space usage alerts that you can enable: **warning**, for when tablespace space is somewhat low, and **critical**, for when the tablespace is almost completely full and action must be taken immediately.

For both warning and critical alerts, you can specify alert thresholds in the following ways:

- By space used (%)

When space used becomes greater than or equal to a percentage of total space, an alert is issued.

- By free space (MB)

When remaining space falls below an amount (in MB), an alert is issued.

Free-space thresholds are more useful for large tablespaces. For example, for a 10 TB tablespace, setting the percentage full critical alert to as high as 99 percent means that the database would issue an alert when there is still 100 GB of free space remaining. Usually, 100 GB remaining would not be a critical situation, and the alert would not be useful. For this tablespace, it might be better to use a free-space threshold, which you could set to issue a critical alert when 5 GB of free space remains.

For both warning and critical alerts for a tablespace, you can enable either the space used threshold or the free-space threshold, or you can enable both thresholds.

To change space usage alert thresholds for tablespaces:

1. Go to the Database Home page.

See "[Accessing the Database Home Page](#)" on page 3-6.

2. At the top of the page, click **Server** to view the Server subpage.

3. In the Storage section, click **Tablespaces**.

The Tablespaces page appears.

4. Select the tablespace whose threshold you want to change, and then click **Edit**.

The Edit Tablespace page appears, showing the General subpage.

5. Click the **Thresholds** link at the top of the page to display the Thresholds subpage.

The screenshot shows the Oracle Database Thresholds subpage. At the top, there are tabs for 'General', 'Storage', and 'Thresholds'. Below the tabs, there are two columns of metrics: 'Available Space (MB) 32,777.98' and 'Space Used (MB) 3.06', and 'Space Used (%) 0.01' and 'Available Free Space (MB) 32,774.92'. Below these metrics is a section titled 'Tablespace Full Metric Thresholds' with the instruction 'Monitor the fullness of the tablespace using either of the metrics below.' There are two main sections: 'Space Used (%)' and 'Free Space (MB)'. Each section has a description of the metric and a 'Modify' button. Under 'Space Used (%)', there are three radio button options: 'Use Database Default Thresholds' (selected), 'Specify Thresholds', and 'Disable Thresholds'. The 'Specify Thresholds' option has input fields for 'Warning (%)' (85) and 'Critical (%)' (97). Under 'Free Space (MB)', there are three radio button options: 'Use Database Default Thresholds' (selected), 'Specify Thresholds', and 'Disable Thresholds'. The 'Specify Thresholds' option has input fields for 'Warning (MB)' and 'Critical (MB)', both of which are currently empty.

6. In the Space Used (%) section, do one of the following:

- Accept the default thresholds.
- Select **Specify Thresholds**, and then enter a **Warning (%)** threshold and a **Critical (%)** threshold.
- Select **Disable Thresholds** to disable the percentage full thresholds.

7. In the Free Space (MB) section, do one of the following:

- Accept the default thresholds.
 - Select **Specify Thresholds**, and then enter a **Warning (MB)** threshold and a **Critical (MB)** threshold.
 - Select **Disable Thresholds** to disable the threshold for free space remaining.
8. Click **Apply**.
- A confirmation message appears.

Taking a Tablespace Offline

You may want to take a tablespace offline for any of the following reasons:

- To make a portion of the database unavailable while still allowing access to the remainder of the database
- To make an application and its group of tables temporarily unavailable while updating or maintaining the application
- To perform an offline tablespace backup (even though a tablespace can be backed up while online and in use)
- To recover a tablespace after a hardware or software failure
- To rename or relocate tablespace data files

To take a tablespace offline:

1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Storage section, click **Tablespaces**.
The Tablespaces page appears.
4. Select the tablespace to take offline, and then click **Edit**.
The Edit Tablespace page appears.
5. In the Status section, select **Offline**.
For more information about the selections on this page, click **Help**.
6. Make a selection from the **Offline Mode** list, and then click **Apply**.
A confirmation message appears.

Note: To bring the tablespace back online, return to this page, and then, under the Status section, click **Read Write**.

See Also:

- *Oracle Database Administrator's Guide* for more information about taking tablespaces offline and for information about renaming or relocating data files.

Dropping a Tablespace

After a tablespace has been dropped (deleted), 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.

To drop a tablespace:

1. Go to the Database Home page.

See "[Accessing the Database Home Page](#)" on page 3-6.

2. At the top of the page, click **Server** to view the Server subpage.

3. In the Storage section, click **Tablespaces**.

The Tablespaces page appears.

4. Select the tablespace to drop, and then click **Delete**.

Database Control asks for confirmation to delete the tablespace and gives you the option to also delete the associated data files from the disk.

5. Click **Yes** to remove the tablespace.

A confirmation is displayed and the deleted tablespace no longer appears on the Tablespaces page.

See Also:

- "[About Tablespaces](#)" on page 6-7

Reclaiming Unused Space

This section provides background information about reclaimable unused space in the database, and provides instructions about how to reclaim this space. It contains the following topics:

- [About Reclaimable Unused Space](#)
- [About the Segment Advisor](#)
- [Viewing Segment Advisor Recommendations and Reclaiming Space](#)
- [Running the Segment Advisor Manually](#)

About Reclaimable Unused Space

Over time, performing insert, update, and delete operations (also referred to as DML operations) on objects within a tablespace can create pockets of empty space that individually are not big enough to be reused. Collectively, these pockets can waste large amounts of space. The sparsely populated objects that result can degrade performance during queries and DML operations.

Oracle Database enables you reclaim this empty space with the following online operations—that is, operations that do not block queries or DML against affected objects while the operations are proceeding:

- Online segment shrink

Online segment shrink operates on table segments and on the segments of the dependent objects of the table, such as indexes and partitions. For each segment, data is compacted to the front of the segment. Free space can either be returned to the tablespace or kept in the segment for future insert operations.

Online segment shrink is permitted only on segments in a locally managed tablespace with automatic segment space management. Online segment shrink is an in-place operation; no additional free space in the tablespace is required.

- **Online table redefinition (also known as *reorganization*)**

Reorganization relocates the table and its dependent objects in a different part of the tablespace. This operation has the desirable side effect of compacting the data for those objects. For this operation to succeed, however, the tablespace must have free space equal to the size of the table and its dependent objects. Reorganization is the only permitted operation for dictionary-managed tablespaces or for locally managed tablespaces with manual segment space management.

The Segment Advisor generates recommendations for shrinking or reorganizing segments that have a significant amount of reclaimable unused space.

See Also:

- *Oracle Database Administrator's Guide* for more information about online segment shrink and online redefinition of tables

About the Segment Advisor

The Segment Advisor identifies database objects that have unused space that you can reclaim. It performs its analysis by examining usage and growth statistics and by sampling the data in the object. By default, it is configured to run automatically at regular intervals, during all maintenance windows (time periods). You can also run the Segment Advisor manually.

The regularly scheduled Segment Advisor task is known as the Automatic Segment Advisor. Results from the advisor are summarized on the Space Summary section of the Database Home page as Segment Advisor Recommendations.

During each run of the Automatic Segment Advisor, only a subset of the segments in the database are analyzed. If you suspect that segments have reclaimable space, but these segments do not appear among the Segment Advisor recommendations, then the Automatic Segment Advisor may not have chosen them for analysis. You can run the Segment Advisor manually on those segments at any time. See *Oracle Database Administrator's Guide* for information about how the Automatic Segment Advisor selects tablespaces and segments for analysis.

Segment Advisor Recommendations Segment Advisor recommendations are grouped by tablespace. Only tablespace names appear on the Segment Advisor Recommendations page in Database Control.

If a tablespace contains any segments for which a shrink operation or reorganization is recommended, then the tablespace appears on the Segment Advisor Recommendations page. This page estimates the amount of reclaimable space for each tablespace. If you request recommendation details for a tablespace, then Database Control displays the segments in that tablespace for which recommendations were generated.

The recommendations provided by the Segment Adviser depend on the characteristics of the tablespace in which the segment is located. If the tablespace was created with default options, then the Segment Advisor typically recommends shrinking. If the segment is not eligible for online segment shrinking, or if it is in a dictionary-managed tablespace or a tablespace created with manual segment space management, then the Segment Advisor recommends reorganization instead.

In addition, the Automatic Segment Advisor evaluates tables that are 10MB or larger and have at least three indexes to determine the amount of space that could be saved if the tables are compressed with the OLTP compression method.

See Also:

- *Oracle Database Administrator's Guide* for more information about maintenance windows

Viewing Segment Advisor Recommendations and Reclaiming Space

Each Segment Advisor recommendation includes a button that you can click to begin the space reclamation process. Although space reclamation is an online process, it can consume significant resources, so you should reclaim space during off-peak hours.

To view Segment Advisor recommendations and reclaim space:

1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. In the Space Summary section, click the numeric link adjacent to Segment Advisor Recommendations.

Space Summary	
Database Size (GB)	2.52
Problem Tablespaces	2
Segment Advisor Recommendations	9
Policy Violations	0
Dump Area Used (%)	75

The Segment Advisor Recommendations page appears, showing recommendations from the most recent automatic and manual runs of the Segment Advisor.

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. Oracle recommends shrinking, reorganizing or compressing these segments to release unused space. Select the Recommendation Details button to view and implement the recommendations.							
View: All Recommendations ▾							
Recommendation Details							
Select	Tablespace	Recommendations	Tablespace Size (MB)	Evaluated Space (%)	Reclaimable Space (MB) ▾	Extent Management	Segment Space Management
<input checked="" type="radio"/>	TEST_ASSM	4	500.00	87.41	433.98	LOCAL	AUTO
<input type="radio"/>	TEST_MSSM_NEW	2	500.00	86.54	431.00	LOCAL	MANUAL
<input type="radio"/>	TEST_ASSM_NEW	3	500.00	56.58	281.20	LOCAL	AUTO

3. (Optional) In the View list, select **Recommendations for Last Automatic Run** to view recommendations from only the most recent automatic run of the Segment Advisor.
4. Select a tablespace, and then click **Recommendation Details**.

The Recommendation Details for Tablespace page appears.

This page lists the tablespace segments for which online segment shrink or reorganization is recommended.

Recommendation Details for Tablespace: TEST_ASSM_NEW

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. Oracle recommends shrinking, reorganizing or compressing these segments to release unused space. Select the segment to implement the recommendation.

Schema Recommendation **All Types**

Segment Minimum Reclaimable Space (MB)

Partition

Select	Schema	Segment	Recommendation	Reclaimable Space (MB) ▾	Allocated Space (MB)	Used Space (MB)	Segment Type
<input type="checkbox"/>	SYSTEM	TEST_SHRINK_TABLE_NEW	<input type="button" value="Shrink"/>	148.20	149.00	0.80	TABLE
<input type="checkbox"/>	SYSTEM	TEST_SHRINK_NEW_INDEX	<input type="button" value="Shrink"/>	98.00	98.00	0.00	INDEX
<input type="checkbox"/>	SYSTEM	TEST_SHRINK_NEW_INDEX_AUTO	<input type="button" value="Shrink"/>	35.00	35.00	0.00	INDEX

5. (Optional) Reduce the number of segments displayed in the segment list by entering search criteria in the Schema, Segment, Partition, or Minimum Reclaimable Space fields, and then clicking **Search**.
6. Examine the recommendation for each segment, and then proceed as follows:
 - If the recommendation for any segments is to reorganize, then start reorganization for those segments by clicking the **Reorganize** button, or select multiple segments and then click the Reorganize button above the table. Follow the directions in the succeeding pages to generate a reorganization script for the selected segments.
 - If the recommendation for any segments is to shrink, then proceed with Step 7 through Step 13 for those segments.
7. Do one of the following to select one or more segments to shrink:
 - To reclaim space in a single segment, click **Shrink** under the Recommendation column for that segment.
 - To shrink one or more segments, select the segments, and then click the **Shrink** button above the table.
8. On the Shrink Segment: Options page, accept **Compact Segments and Release Space**.

This option returns freed space to the tablespace. If you do not want to release the freed space to the tablespace, then choose **Compact Segments**. You can rerun the shrink process later to release the freed space.

Note: Choose **Compact Segments** if you believe that you may have long-running queries currently in operation that involve the segments being shrunk. Long-running queries that were started before the shrink operation completes might attempt to read from blocks that have been reclaimed, which results in an error. Alternatively, you can schedule the shrink operation for an off-peak period in Step 10.

9. Click **Implement**.
The Shrink Segment: Schedule page appears.

10. Note your job name, or replace it with a job name of your choosing.
11. Under the heading Start, select **Immediately**.
Because the shrink operation can be resource intensive, you can also select **Later** and schedule the operation for an off-peak period.
12. Click **Submit**.
The Scheduler Jobs page appears, and shows the shrink job in the Running subpage.
13. Click **Refresh** to update the page until the job disappears from the Running subpage.
You can then switch to the History or All subpage to view the completed status of the job.

Running the Segment Advisor Manually

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

See *Oracle Database Administrator's Guide* for information about running the Segment Advisor manually.

Managing the Online Redo Log

The online redo log files are a critical component in database recovery. Every transaction in the database updates the redo logs, regardless of whether archiving is enabled. During crash, instance, or media recovery, the database properly applies redo log files in ascending order by using the log sequence number of the necessary archived and redo log files.

If properly configured, the online redo logs require little maintenance. This section describes the more common redo log management tasks. It contains the following topics:

- [Multiplexing the Online Redo Log](#)
- [Switching a Log File](#)

See Also:

- ["About Online Redo Log Files"](#) on page 6-4
- ["Viewing Online Redo Log File Information"](#) on page 6-12

Multiplexing the Online Redo Log

Oracle recommends that you multiplex the online redo log. Multiplexing provides better protection for data if an instance or media failure occurs.

To multiplex your online redo log, you must add members to each online redo log group. It is not required that online 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 online redo log groups.

Note: When you multiplex the online redo log, the database must increase the amount of I/O that it performs. Depending on your configuration, this action may affect overall database performance.

To multiplex the online redo log:

1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Storage section, click **Redo Log Groups**.
The Redo Log Groups page appears.
4. Select a group and click **Edit**.
The Edit Redo Log Group page appears.
5. Click **Add**.
The Add Redo Log Member page appears.
6. If the Storage Type is set to File System, then in the **File Name** field, enter a 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.
7. In the **File Directory** field, enter the directory where you want the data file to be stored on disk.

You can create this file in the same directory as the other member of the redo log file group, but it is recommended that you store members on separate disk drives. That way, if there is a drive failure, then you still have access to one member.
8. Click **Continue**, and then click **Apply** to accept your changes.

A confirmation message appears.
9. At the top of the page, click the **Redo Log Groups** link to return to the Redo Log Groups page.
10. Repeat Step 4 through Step 9 for every existing log group.

Switching a Log File

When a log switch occurs, the log writer (LGWR) process stops writing to the current online redo log group and starts writing to the next available redo log group. After a log switch, the current online redo log group becomes inactive, and the next available online redo log group becomes the current online redo log group.

You can force a log switch to make the current redo group inactive and available for redo log maintenance operations. Forcing a log switch is useful in configurations with large redo log files that take a long time to fill. For example, you might want to:

- Drop the current redo group, but are not able to do so until the group is inactive
- Archive the current online redo log group members immediately, even though they are not yet completely filled

To switch a log file:

1. Go to the Database Home page.

See "[Accessing the Database Home Page](#)" on page 3-6.

2. At the top of the page, click **Server** to view the Server subpage.
3. In the Storage section, click **Redo Log Groups**.

The Redo Log Groups page appears.

4. In the Actions list, select **Switch logfile**, and then click **Go**.

A confirmation message appears. The status of the group that had been Current changes to Active, and the status of the next group in the list changes from Inactive to Current.

Managing Undo Data

Beginning with Oracle Database 11g, for a default installation, Oracle Database automatically manages the undo data. There is typically no need for database administrator (DBA) intervention. However, if your installation uses Oracle Flashback operations, then you may have to perform some undo management tasks to ensure the success of these operations.

This section provides background information and instructions for managing undo data. It contains the following topics:

- [About Undo Data](#)
- [About Managing Undo Data](#)
- [Viewing Undo Information](#)
- [Computing the Minimum Undo Tablespace Size Using the Undo Advisor](#)
- [Changing the Undo Tablespace to a Fixed Size](#)

About Undo Data

When a transaction modifies data, Oracle Database 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 if a rollback is necessary. A rollback can be needed because a user 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 changes may be occurring against the data. With read consistency, a user session does not see uncommitted changes made in other user sessions (sometimes referred to as *dirty reads*). For example, if a user issues a query at 10:00 a.m. and the query lasts for 15 minutes, then the query results reflect the entire state of the data at 10:00 a.m., regardless of update or insert operations performed by other users after the query started.
- To enable certain Oracle Flashback features, such as Oracle Flashback Query and Oracle Flashback Table, which enable you to view or recover data to a previous point in time.

Undo Tablespace

With automatic undo management, undo data is stored in an undo tablespace. Undo tablespaces have additional properties beyond those of permanent tablespaces. There can be multiple undo tablespaces, but only one can be active for an Oracle instance.

When you create the database using Database Configuration Assistant (DBCA), it creates an autoextending undo tablespace named `UNDOTBS1`, with a maximum extension size of 32,767 MB.

Undo Retention

Oracle Database automatically ensures that undo data that is in use by an active transaction is never overwritten until that transaction has been committed. After the transaction has been committed, the space occupied by that undo data can be reused, or overwritten. In this case, that undo data could be overwritten if space in the undo tablespace becomes scarce.

Even after a transaction has been committed, it is useful to retain (not overwrite) its undo data, to ensure the success of Oracle Flashback features and for read consistency for long-running queries. To this end, the database maintains and automatically tunes an **undo retention period**. Committed undo data whose age is less than the undo retention period is retained for use by queries or Oracle Flashback operations.

See Also:

- *Oracle Database Concepts* for more information about read consistency
- *Oracle Database Advanced Application Developer's Guide* for more information about Oracle Flashback features

About Managing Undo Data

Although by default Oracle Database manages undo data and the undo tablespace automatically, if your installation uses Oracle Flashback features, then you may have to perform some undo management tasks to ensure the success of these operations.

Oracle Flashback operations resulting in `snapshot too old` errors indicate that you must intervene to ensure that sufficient undo data is retained to support these operations.

The following methods better support Oracle Flashback operations:

- Set the minimum undo retention period for the autoextending tablespace to be as long as the longest expected Oracle Flashback operation.

You achieve this goal by setting the `UNDO_RETENTION` initialization parameter. See *Oracle Database Administrator's Guide* for details.

- Change the undo tablespace to a fixed size.

For an autoextending undo tablespace, Oracle Database always automatically tunes the undo retention period to be slightly longer than the longest-running active query. However, this autotuned retention period may be insufficient to accommodate Oracle Flashback operations. If the undo tablespace has autoextending disabled, or has a fixed size, then Oracle Database uses a different method for tuning the undo retention period to better accommodate Oracle Flashback operations.

To change the undo tablespace to a fixed size, you must choose a tablespace size that is sufficiently large. If you choose an undo tablespace size that is too small, then the following errors could occur:

- DML could fail because there is not enough space to accommodate undo data for new transactions.

- Long-running queries could fail with a `snapshot too old` error, which means that there was insufficient undo data for read consistency.

Oracle Enterprise Manager Database Control (Database Control) includes an Undo Advisor to help you determine the minimum size for the fixed size of the undo tablespace. See ["Computing the Minimum Undo Tablespace Size Using the Undo Advisor"](#) on page 6-34.

See Also:

- ["About Undo Data"](#) on page 6-30
- ["Computing the Minimum Undo Tablespace Size Using the Undo Advisor"](#) on page 6-34
- ["Changing the Undo Tablespace to a Fixed Size"](#) on page 6-37

Viewing Undo Information

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

- Name and current size of the undo tablespace
- Autoextend tablespace setting (Yes or No)
- Current autotuned undo retention period

To view undo information:

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Server** to view the Server subpage.
3. In the Database Configuration section, click **Automatic Undo Management**.
The Automatic Undo Management page appears.

Automatic Undo Management

In the General tab, you can view the current undo settings for your instance and use the Undo Advisor to analyze the undo tablespace requirements. This analysis can be performed based on the specified analysis period or the desired undo retention. The system activity for the specified time period can be viewed in the System Activity tab.

General System Activity

Undo Retention Settings

Undo Retention (minutes) **180**
Retention Guarantee No

Undo Tablespace for this Instance

Tablespace UNDOTBS [Change Tablespace](#)
Size (MB) **234**
Auto-Extensible Yes

Undo Advisor: Undo Retention and Undo Tablespace Sizing Advice

Undo retention is the length of time that undo data is retained in the undo tablespaces. Undo data must be retained for the length of the longest running query, the longest running transaction, and the longest flashback duration (except for Flashback Database). The undo tablespace should be sized large enough to hold the undo generated by the database during the undo retention period. Note that the undo retention parameter is also used as the retention value for LOB columns.

Analysis Period

Analysis Time Period Last Seven Days ▼
 Desired Undo Retention Automatically chosen based on longest query in analysis period
 Specified manually to allow for longer duration queries or flashback
 Duration minutes ▼
[Run Analysis](#)

Analysis Results

[Edit Undo Tablespace](#) [Edit Undo Retention](#)

Selected Analysis Time Period **Jun 12, 2009 3:00:00 PM PDT To Jun 19, 2009 3:00:00 PM PDT**
 Minimum Required Undo Tablespace Size (MB) **274**
 Recommended Undo Tablespace Size (MB) **274**
 TIP Recommended size is three times the minimum size to allow for workload fluctuations

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

[Show Graph](#)

See Also:

- ["About Undo Data"](#) on page 6-30

Computing the Minimum Undo Tablespace Size Using the Undo Advisor

If you must change the undo tablespace to a fixed size, then use the Undo Advisor to help determine the minimum required size. You can also use the Undo Advisor to set the minimum undo retention period.

To configure the undo tablespace to have a fixed size, Oracle suggests that you first allow enough time after database creation to run a full workload, thus allowing the undo tablespace to grow to its minimum required size to handle the workload. Then, you can use the Undo Advisor to determine the best size to configure the undo tablespace to allow for future long-running queries and Oracle Flashback operations.

To compute the minimum undo tablespace size using the Undo Advisor:

1. Go to the Automatic Undo Management page.
 See ["Viewing Undo Information"](#) on page 6-32.
2. Decide whether you want to compute the minimum size of the undo tablespace based on statistics gathered over a designated time period (such as the last 7 days), or based on an undo retention period of a duration that you choose.

The automatically gathered statistics include the duration of the longest-running query and the undo generation rate. Computing the minimum undo tablespace size based on these statistics is advisable if you do not use Oracle Flashback

features or if you do not expect future long-running queries to exceed the duration of previous long-running queries.

If you prefer to choose and enter the duration of a desired undo retention period, then the duration must be based on your expectations of the duration of future long-running queries or Oracle Flashback operations.

3. To compute the minimum undo tablespace size based on statistics gathered over a period, complete the following steps:

- a. In the Analysis Period section, in the Analysis Time Period list, select the desired analysis time period.

If you select **Customize Time Period**, then a page appears that enables you to enter the starting and ending date and time for the period.

- b. Select **Automatically chosen based on longest query in analysis period**.
- c. Click **Run Analysis**.

The minimum required undo tablespace size is displayed in the Analysis Results section.

- d. (Optional) Click **Show Graph** to view a graph of the analysis results.

4. To compute the minimum undo tablespace size based on a duration that you enter, complete the following steps:

- a. In the Analysis Period section, select **Specified manually to allow for longer duration queries or flashback**.

- b. In the **Duration** field, enter the desired duration of the undo retention period in seconds, minutes, hours, or days.

You may compute this duration as follows:

- Determine the duration of the expected longest-running query for your workload.

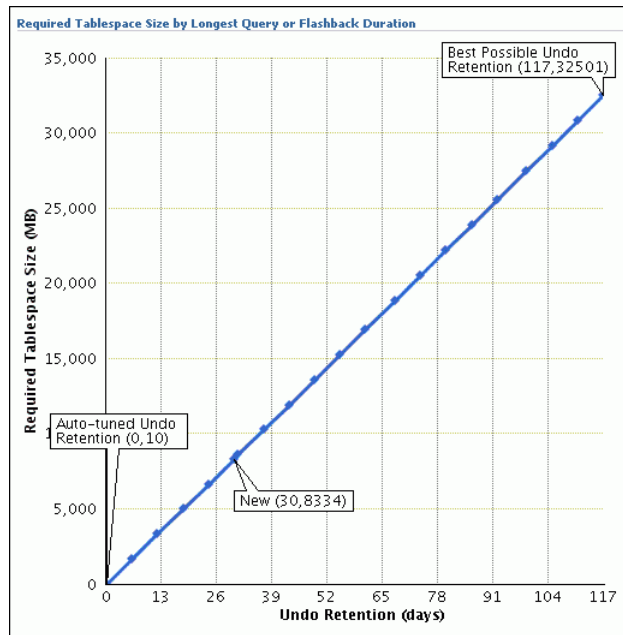
You can base your determination on the longest-running query recorded during a previous workload period. This information is available from the System Activity subpage of the Automatic Undo Management page. You choose the analysis period for this subpage by selecting from the Analysis Time Period list on the General subpage. You can choose a custom analysis period that corresponds to your typical workload period.

- Determine the longest duration that is expected for an Oracle Flashback operation.
- Take the maximum of these two durations.

- c. In the Analysis Period section of the General subpage, click **Run Analysis**.

The minimum required undo tablespace size is displayed in the Analysis Results section.

- d. (Optional) Click **Show Graph** to view a graph of the analysis results.



- (Optional) Click the tick-mark, or point, on the curve that corresponds to the desired undo retention period.

The Duration field changes to match the selected undo retention period, and the Minimum Required Undo Tablespace Size field above the graph changes to reflect the matching required size.

- (Optional) Click **Apply**.

The minimum undo retention period is set to the value of the Duration field.

Note: Running an analysis or setting the minimum undo retention with the Undo Advisor does not change the size of the undo tablespace. You must follow the instructions in "[Changing the Undo Tablespace to a Fixed Size](#)" on page 6-37 to change the size of the undo tablespace.

See Also:

- "[About Undo Data](#)" on page 6-30
- "[About Managing Undo Data](#)" on page 6-31

Changing the Undo Tablespace to a Fixed Size

You change the undo tablespace to a fixed size to prevent the tablespace from growing too large or to better support Oracle Flashback operations.

To change the undo tablespace to a fixed size:

- On the Automatic Undo Management page, after determining the minimum required undo tablespace size, click **Edit Undo Tablespace**.

The Edit Tablespace page appears, displaying the properties of the undo tablespace.

- In the Datafiles section, click **Edit**.

The Edit Datafile page appears.

3. In the **File Size** field, enter the computed minimum size for the undo tablespace. See "[Computing the Minimum Undo Tablespace Size Using the Undo Advisor](#)" on page 6-34.
4. In the Storage section, deselect **Automatically extend data file when full (AUTOEXTEND)**.
5. Click **Continue**.

The Edit Tablespace page returns.

6. Click **Apply**.

A confirmation message appears.

See Also:

- "[About Undo Data](#)" on page 6-30
- "[About Managing Undo Data](#)" on page 6-31

Storage: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Storage OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/storage/storage.htm

Administering User Accounts and Security

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

- [About User Accounts](#)
- [About User Privileges and Roles](#)
- [About Administrative Accounts and Privileges](#)
- [Administering Roles](#)
- [Administering Database User Accounts](#)
- [Setting the Database Password Policy](#)
- [Users: Oracle By Example Series](#)

About User Accounts

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

- Authentication method
- Password for database authentication
- Default tablespaces for permanent and temporary data storage
- Tablespace quotas
- Account status (locked or unlocked)
- Password status (expired or not)

When you create a user account, you must not only assign a user name, a password, and default tablespaces for the account, but you must also do the following:

- Grant the appropriate system privileges, object privileges, and roles to the account.
- If the user will be creating database objects, then give the user account a space usage quota on each tablespace in which the objects will be created.

Oracle recommends that you grant each user just enough privileges to perform his job, and no more. For example, a database application developer needs privileges to create and modify tables, indexes, views, and stored procedures, but does not need (and should not be granted) privileges to drop (delete) tablespaces or recover the database. You can create user accounts for database administration, and grant only a subset of administrative privileges to those accounts.

In addition, you may want to create user accounts that are used by applications only. That is, nobody logs in with these accounts; instead, applications use these accounts to connect to the database, and users log in to the applications. This type of user account avoids giving application users the ability to log in to the database directly, where they could unintentionally cause damage. See ["About User Privileges and Roles"](#) on page 7-3 for more information.

When you create a user account, you are also implicitly creating a schema for that user. A **schema** is a logical container for the database objects (such as tables, views, triggers, and so on) that the user creates. The schema name is the same as the user name, and can be used to unambiguously refer to objects owned by the user. For example, `hr.employees` refers to the table named `employees` in the `hr` schema. (The `employees` table is owned by `hr`.) The terms *database object* and *schema object* are used interchangeably.

When you delete a user, you must either simultaneously delete all schema objects of that user, or you must have previously deleted the schema objects in separate operations.

Predefined User Accounts

In addition to the user accounts that you create, the database includes several user accounts that are automatically created upon installation.

All databases include the administrative accounts `SYS`, `SYSTEM`, `SYSMAN`, and `DBSNMP`. **Administrative accounts** are highly privileged accounts, and are needed only by individuals authorized to perform administrative tasks such as starting and stopping the database, managing database memory and storage, creating and managing database users, and so on. You log in to Oracle Enterprise Manager Database Control (Database Control) with `SYS`, `SYSTEM`, or `SYSMAN`. The Management Agent of Database Control uses the `DBSNMP` account to monitor and manage the database. You assign the passwords for these accounts when you create the database with Oracle Database Configuration Assistant (DBCA). You must not delete these accounts.

All databases also include **internal accounts**, which are automatically created so that individual Oracle Database features or components such as Oracle Application Express can have their own schemas. To protect these accounts from unauthorized access, they are initially locked and their passwords are expired. (A **locked account** is an account for which login is disabled.) You must not delete internal accounts, and you must not use them to log in to the database.

Your database may also include **sample schemas**, which are a set of interlinked schemas that enable Oracle documentation and Oracle instructional materials to illustrate common database tasks. These schemas also provide a way for you to experiment without endangering production data.

Each sample schema has a user account associated with it. For example, the `hr` user account owns the `hr` schema, which contains a set of simple tables for a human resources application. The sample schema accounts are also initially locked and have an expired password. As the database administrator, you are responsible for unlocking these accounts and assigning passwords to these accounts.

See Also:

- *Oracle Database 2 Day + Security Guide* for a list of predefined user accounts
- ["Locking and Unlocking User Accounts"](#) on page 7-21
- ["About Administrative Accounts and Privileges"](#) on page 7-4
- ["Administering Database User Accounts"](#) on page 7-11
- *Oracle Database Sample Schemas* for a description of the sample schemas
- *Oracle Database Concepts* for an overview of database security

About User Privileges and Roles

User privileges 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. When creating a user, you grant privileges to enable the user to connect to the database, to run queries and make updates, to create schema objects, and more.

The main types of user privileges are as follows:

- **System privileges**—A system privilege gives a user the ability to perform a particular action, or to perform an action on any schema objects of a particular type. For example, the system privilege `CREATE TABLE` permits a user to create tables in the schema associated with that user, and the system privilege `CREATE USER` permits a user to create database users.
- **Object privileges**—An object privilege gives a user the ability to perform a particular action on a specific schema object. Different object privileges are available for different types of schema objects. The privilege to select rows from the `EMPLOYEES` table or to delete rows from the `DEPARTMENTS` table are examples of object privileges.

Managing privileges is made easier by using **roles**, which are named groups of related privileges. You create roles, grant system and object privileges to the roles, and then grant roles to users. You can also grant roles to other roles. Unlike schema objects, roles are not contained in any schema.

[Table 7-1](#) lists three widely used roles that are predefined in Oracle Database. You can grant these roles when you create a user or at any time thereafter.

Table 7-1 Oracle Database Predefined Roles

Role Name	Description
CONNECT	Enables a user to connect to the database. Grant this role to any user or application that needs database access. If you create a user using Database Control, then this role is automatically granted to the user.

Table 7-1 (Cont.) Oracle Database Predefined Roles

Role Name	Description
RESOURCE	Enables a user to create, modify, and delete certain types of schema objects in the schema associated with that user. Grant this role only to developers and to other users that must create schema objects. This role grants a subset of the create object system privileges. For example, it grants the CREATE TABLE system privilege, but does not grant the CREATE VIEW system privilege. It grants only the following privileges: CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE.
DBA	Enables a user to perform most administrative functions, including creating users and granting privileges; creating and granting roles; creating, modifying, and deleting schema objects in any schema; and more. It grants all system privileges, but does not include the privileges to start or shut down the database instance. It is by default granted to users SYS and SYSTEM.

See Also:

- ["Administering Roles"](#) on page 7-6
- ["Administering Database User Accounts"](#) on page 7-11
- [Chapter 8, "Managing Schema Objects"](#)
- *Oracle Database 2 Day + Security Guide* for more information about privileges and roles
- *Oracle Database SQL Language Reference* for tables of system privileges, object privileges, and predefined roles
- *Oracle Database Concepts* for an overview of database security

About Administrative Accounts and Privileges

Administrative accounts and privileges enable you to perform administrative functions such as managing users, managing database memory, and starting up and shutting down the database.

This section contains the following topics:

- [SYS and SYSTEM Users](#)
- [SYSDBA and SYSOPER System Privileges](#)

See Also:

- ["About User Accounts"](#) on page 7-1
- ["About User Privileges and Roles"](#) on page 7-3
- ["Administering Database User Accounts"](#) on page 7-11

SYS and SYSTEM Users

The following administrative user accounts are automatically created when you install Oracle Database. They are both created with the password that you supplied upon installation, and they are both automatically granted the DBA role.

- **SYS**

This account can perform all administrative functions. All base (underlying) tables and views for the database data dictionary are stored in the *SYS* schema. 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. You must not create any tables in the *SYS* schema.

The *SYS* user is granted the *SYSDBA* privilege, which enables a user to perform high-level administrative tasks such as backup and recovery.

- **SYSTEM**

This account can perform all administrative functions except the following:

- Backup and recovery
- Database upgrade

While this account can be used to perform day-to-day administrative tasks, Oracle strongly recommends creating named users account for administering the Oracle database to enable monitoring of database activity.

SYSDBA and SYSOPER System Privileges

SYSDBA and *SYSOPER* are administrative privileges required to perform high-level administrative operations such as creating, starting up, shutting down, backing up, or recovering the database. The *SYSDBA* system privilege is for fully empowered database administrators and the *SYSOPER* system privilege allows a user to perform basic operational tasks, but without the ability to look at user data.

The *SYSDBA* and *SYSOPER* system privileges allow access to a database instance even when the database is not open. Control of these privileges is therefore completely outside of the database itself. This control enables an administrator who is granted one of these privileges to connect to the database instance to start the database.

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 using `AS SYSDBA`.

The *SYS* user is automatically granted the *SYSDBA* privilege upon installation. When you log in as user *SYS*, you must connect to the database as *SYSDBA* or *SYSOPER*. Connecting as a *SYSDBA* user invokes the *SYSDBA* privilege; connecting as *SYSOPER* invokes the *SYSOPER* privilege. Oracle Enterprise Manager Database Control does not permit you to log in as user *SYS* without connecting as *SYSDBA* or *SYSOPER*.

When you connect with the *SYSDBA* or *SYSOPER* privilege, you connect with a default schema, not with the schema that is generally associated with your user name. For *SYSDBA* this schema is *SYS*; for *SYSOPER* the schema is *PUBLIC*.

Caution: When you connect as user *SYS*, you have unlimited privileges on data dictionary tables. Be certain that you do not modify any data dictionary tables.

See Also:

- *Oracle Database Administrator's Guide* for the operations authorized with the *SYSDBA* and *SYSOPER* privileges

Administering Roles

Roles are named groups of related system and object privileges. You create roles and then assign them to users and to other roles.

This section contains the following topics:

- [Viewing Roles](#)
- [Example: Creating a Role](#)
- [Example: Modifying a Role](#)
- [Deleting a Role](#)

See Also:

- ["About User Privileges and Roles"](#) on page 7-3
- *Oracle Database 2 Day + Security Guide* for more information about administering user security, roles, and privileges

Viewing Roles

You view roles on the Roles page of Oracle Enterprise Manager Database Control (Database Control).

To view roles:

1. Go to the Database Home page, logging in with a user account that has privileges to manage roles. An example of such a user account is `SYSTEM`.

See ["Accessing the Database Home Page"](#) on page 3-6.

2. At the top of the page, click **Server** to view the Server subpage.
3. In the Security section, click **Roles**.

The Roles page appears.

Select	Role	Authentication
<input checked="" type="radio"/>	ADM_PARALLEL_EXECUTE_TASK	NO
<input type="radio"/>	AQ_ADMINISTRATOR_ROLE	NO
<input type="radio"/>	AQ_USER_ROLE	NO
<input type="radio"/>	AUTHENTICATEDUSER	NO
<input type="radio"/>	CONNECT	NO
<input type="radio"/>	CSW_USR_ROLE	NO
<input type="radio"/>	CTXAPP	NO
<input type="radio"/>	CWM_USER	NO
<input type="radio"/>	DATAPUMP_EXP_FULL_DATABASE	NO
<input type="radio"/>	DATAPUMP_IMP_FULL_DATABASE	NO

4. To view the details of a particular role, in the **Select** column, select the name of the role you want to view, and then click **View**.

If you do not see the role, then it may be on another page. In this case, do one of the following:

- Just above the list of roles, click **Next** to view the next page. Continue clicking **Next** until you see the desired role.
- Use the Search area of the page to search for the desired role. In the **Object Name** field, enter the first few letters of the role, and then click **Go**.

You can then select the role and click **View**.

The View Role page appears. In this page, you can see all the privileges and roles granted to the selected role.

Example: Creating a Role

Suppose you want to create a role called APPDEV for application developers. Because application developers must be able to create, modify, and delete the schema objects that their applications use, you want the APPDEV role to include the system privileges shown in [Table 7-2](#).

Table 7-2 System Privileges Granted to the APPDEV Role

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

To create the APPDEV role:

1. Go to the Roles page, as described in "[Viewing Roles](#)" on page 7-6.
2. Click **Create**.
The Create Role page appears.
3. In the **Name** field, enter APPDEV.
4. Click **System Privileges** to go to the System Privileges subpage.

Database Instance: orcl.example.com > Roles > Logged in As SYSTEM

Create Role

Show SQL Cancel OK

General Roles **System Privileges** Object Privileges Consumer Group Privileges

Edit List

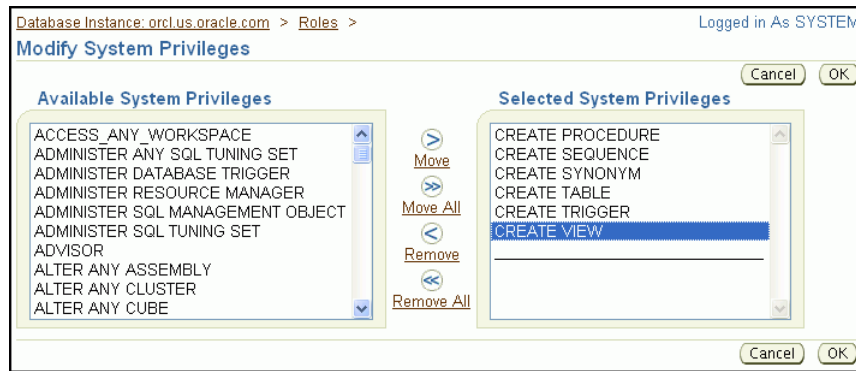
System Privilege	Admin Option
No items found	

General Roles **System Privileges** Object Privileges Consumer Group Privileges

The table of system privileges for this role contains no rows yet.

5. Click **Edit List**.
The Modify System Privileges page appears.
6. In the Available System Privileges list, double-click privileges to add them to the Selected System Privileges list.

The privileges to add are listed in [Table 7-2](#) on page 7-8.



Note: Double-clicking a privilege is a shortcut. You can also select a privilege and then click the **Move** button. To select multiple privileges, hold down the Shift key while selecting a range of privileges, or press the Ctrl key and select individual privileges, then click **Move** after you have selected the privileges.

7. Click **OK**.

The System Privileges subpage returns, showing the system privileges that you selected. At this point, you could click **Roles** to assign other roles to the APPDEV role, or click **Object Privileges** to assign object privileges to the APPDEV role.

8. Click **OK** to return to the Roles page.

The APPDEV role now appears in the table of database roles.

Example: Modifying a Role

Suppose your applications make use of Oracle Streams Advanced Queuing, and you determine that developers must be granted the roles AQ_ADMINISTRATOR_ROLE and AQ_USER_ROLE to develop and test their applications. You must edit the APPDEV role to grant it these two Advanced Queuing roles.

To modify the APPDEV role:

1. Go to the Roles page, as described in "[Viewing Roles](#)" on page 7-6.

2. In the Select column, click **APPDEV** role, and then click **Edit**.

The Edit Role page appears.

3. Click **Roles** to navigate to the Roles subpage.

4. Click **Edit List**.

The Modify Roles page appears.

5. In the Available Roles list, double-click the roles AQ_ADMINISTRATOR_ROLE and AQ_USER_ROLE to add them to the Selected Roles list.

6. Click **OK**.

The Roles subpage returns, showing that the roles that you selected were granted to the APPDEV role.

7. Click **Apply** to save your changes.

An update message appears, indicating that the role APPDEV was modified successfully.

Deleting a Role

Use caution when deleting a role, because Database Control deletes a role even if that role is currently granted to one or more users. Before deleting a role, you may want to determine if the role has any grantees. Dropping (deleting) a role automatically removes the privileges associated with that role from all users that had been granted the role.

To determine if a role has any grantees:

1. Go to the Roles page as described in "[Viewing Roles](#)" on page 7-6.
2. In the Select column, click the desired role.

If you do not see the desired role, then it may be on another page. In this case, do one of the following:

- Just above the list of roles, click **Next** to view the next page. Continue clicking **Next** until you see the desired role.
- Use the Search area of the page to search for the desired role. In the **Object Name** field, enter the first few letters of the role, and then click **Go**.

You can then select the role.

3. In the Actions list, select **Show Grantees**, and then click **Go**.

A report appears, listing the users that are granted the selected role.

4. Click **Cancel** to return to the Roles page.

To delete a role:

1. If you are not there, then go to the Roles page, as described in "[Viewing Roles](#)" on page 7-6.
2. In the Select column, click the desired role, and then click **Delete**.

A confirmation page appears.

3. Click **Yes**.

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

Administering Database User Accounts

This section provides instructions for creating and managing user accounts for the people and applications that use your database. It contains the following topics:

- [Viewing User Accounts](#)
- [Example: Creating a User Account](#)
- [Creating a New User Account by Duplicating an Existing User Account](#)
- [Example: Granting Privileges and Roles to a User Account](#)
- [Example: Assigning a Tablespace Quota to a User Account](#)
- [Example: Modifying a User Account](#)
- [Locking and Unlocking User Accounts](#)
- [Expiring a User Password](#)

- [Example: Deleting a User Account](#)

See Also:

- ["About User Accounts"](#) on page 7-1

Viewing User Accounts

You view user accounts on the Users page of Oracle Enterprise Manager Database Control (Database Control).

To view users:

1. Go to the Database Home page, logging in with a user account that has privileges to manage users, for example, SYSTEM.

See ["Accessing the Database Home Page"](#) on page 3-6.

2. At the top of the page, click **Server** to view the Server subpage.
3. In the Security section, click **Users**.

The Users page appears.

Database Instance: orcl.us.oracle.com > Logged in As SYSTEM

Users Object Type: User

Search
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:

Actions: Previous 1-25 of 40 Next 15

Select	UserName	Account Status	Expiration Date	Default Tablespace	Temporary Tablespace	Profile	Created	User Type
<input checked="" type="radio"/>	ADAMS	OPEN	Oct 27, 2009 2:46:09 PM PDT	USERS	TEMP	DEFAULT	Apr 30, 2009 2:46:09 PM PDT	LOCAL
<input type="radio"/>	ANONYMOUS	EXPIRED & LOCKED	Apr 17, 2009 10:30:01 AM PDT	SYSAUX	TEMP	DEFAULT	Apr 17, 2009 10:09:42 AM PDT	LOCAL
<input type="radio"/>	APEX_PUBLIC_USER	EXPIRED & LOCKED	Apr 17, 2009 10:30:01 AM PDT	USERS	TEMP	DEFAULT	Apr 17, 2009 10:21:43 AM PDT	LOCAL
<input type="radio"/>	APPQOSSYS	EXPIRED & LOCKED		SYSAUX	TEMP	DEFAULT	Apr 17, 2009 10:06:26 AM PDT	LOCAL
<input type="radio"/>	BI	EXPIRED & LOCKED	Apr 25, 2009 12:52:18 AM PDT	USERS	TEMP	DEFAULT	Apr 25, 2009 12:48:44 AM PDT	LOCAL
<input type="radio"/>	BLAKE	OPEN	Oct 27, 2009 2:48:24 PM PDT	USERS	TEMP	DEFAULT	Apr 30, 2009 2:48:24 PM PDT	LOCAL

4. To view the details of a particular user, in the **Select** column, click the user, and then click **View**.

If you do not see the user, then it may be on another page. In this case, do one of the following:

- Just above the list of users, click **Next** to view the next page. Continue clicking **Next** until you see the desired user.
- Use the Search area of the page to search for the desired user. In the **Object Name** field, enter the first few letters of the user name, and then click **Go**.

- Click a table column to change the sort order of the data in the table. For example, to list the users in reverse alphabetical order, click the `UserName` column heading.

You can then select the user and click **View**.

The View User page appears, and displays all user attributes.

Example: Creating a User Account

Suppose you want to create a user account for a database application developer named Nick. Because Nick is a developer, you want to grant him the database privileges and roles that he requires to build and test his applications. You also want to give Nick a 10 MB quota on his default tablespace so that he can create schema objects in that tablespace.

To create the user Nick:

- Go to the Users page, as described in ["Viewing User Accounts"](#) on page 7-12.
- On the Users page, click **Create**.

The Create User page appears, displaying the General subpage.

The screenshot shows the 'Create User' page in Oracle Database. The page is titled 'Create User' and is logged in as 'SYSTEM'. The 'General' subpage is active, showing the following fields and options:

- Name:** A text input field.
- Profile:** A dropdown menu set to 'DEFAULT'.
- Authentication:** A dropdown menu set to 'Password'.
- Enter Password:** A text input field.
- Confirm Password:** A text input field.
- For Password choice, the role is authorized via password.** A checkbox for 'Expire Password now' is unchecked.
- Default Tablespace:** A text input field with a help icon.
- Temporary Tablespace:** A text input field with a help icon.
- Status:** Radio buttons for 'Locked' and 'Unlocked', with 'Unlocked' selected.

At the bottom of the page, there are buttons for 'Show SQL', 'Cancel', and 'OK'.

- In the **Name** field, enter NICK.
- In the Profile list, accept the value DEFAULT.
This setting assigns the default password policy to user Nick.
See ["Setting the Database Password Policy"](#) on page 7-23.
- Accept the default value Password in the Authentication list.
For information about other more advanced methods to authenticate users, see *Oracle Database Security Guide*.
- In the **Enter Password** and **Confirm Password** fields, enter a password that is secure.
Create a password that is secure. See *Oracle Database Security Guide* for more information.

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

8. (Optional) Next to the **Default Tablespace** field, click the flashlight icon, select the **USERS** tablespace, and then click **Select**.

All schema objects that Nick creates will then be created in the **USERS** tablespace unless he specifies otherwise. If you leave the **Default Tablespace** field blank, then Nick is assigned the default tablespace for the database, which is **USERS** in a newly installed database. For more information about the **USERS** tablespace, see "[About Tablespaces](#)" on page 6-7.

9. (Optional) Next to the **Temporary Tablespace** field, click the flashlight icon, select the **TEMP** tablespace, and then click **Select**.

If you leave the **Temporary Tablespace** field blank, then Nick is assigned the default temporary tablespace for the database, which is **TEMP** in a newly installed database. For more information about the **TEMP** tablespace, see "[About Tablespaces](#)" on page 6-7.

10. For the **Status** option, accept the default selection of **Unlocked**.

You can later lock the user account to prevent users from logging in with it. To temporarily deny access to a user account, locking the user account is preferable to deleting it, because deleting it also deletes all schema objects owned by the user.

11. Grant roles, system privileges, and object privileges to the user, as described in "[Example: Granting Privileges and Roles to a User Account](#)" on page 7-16.

Note: Do not click **OK** in Step 13 of "[Example: Granting Privileges and Roles to a User Account](#)" on page 7-16. Instead, skip that step and continue with Step 12 in this procedure.

12. Assign a 10 MB quota on the **USERS** tablespace, as described in "[Example: Assigning a Tablespace Quota to a User Account](#)" on page 7-19.

13. If you did not click **OK** while assigning the tablespace quota (previous step), then click **OK** now to create the user.

See Also:

- "[Creating Database Control Administrative Users](#)" on page 3-8
- *Oracle Database 2 Day + Security Guide*

Creating a New User Account by Duplicating an Existing User Account

To create a user account that is similar in attributes to an existing user account, you can duplicate the existing user account.

To create a new user account by duplicating an existing user account:

1. Go to the **Users** page, as described in "[Viewing User Accounts](#)" on page 7-12.
2. In the **Select** column, click the user to duplicate.
3. In the **Actions** list, select **Create Like**, and then click **Go**.

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

4. Enter a user name and password, modify the user attributes or privileges if desired, and then click **OK** to save the new user.

The Actions list also provides shortcuts for other actions, and provides a way to display the SQL command used to create a user.

Example: Granting Privileges and Roles to a User Account

Suppose you are creating or modifying a user account named Nick. Because Nick is a database application developer, you want to grant him the `APPDEV` role, which enables him to create database objects in his own schema. (You created the `APPDEV` role in "Example: Creating a Role" on page 7-8.) Because you want Nick to be able to create tables and views in other schemas besides his own, you want to grant him the `CREATE ANY TABLE` and `CREATE ANY VIEW` system privileges. In addition, because he is developing a human resources application, you want him to be able to view the tables in the `hr` sample schema and use them as examples. You therefore want to grant him the `SELECT` object privilege on those tables. Finally, you want Nick to be able to log in to Database Control so that he can use the graphical user interface to create and manage his database objects. You therefore want to grant him the `SELECT_CATALOG_ROLE` role. The following table summarizes the privileges and roles to grant to Nick.

Grant Type	Privilege or Role Name
System privileges	<code>CREATE ANY TABLE</code> , <code>CREATE ANY VIEW</code>
Object privileges	<code>SELECT</code> on all tables in the <code>hr</code> schema
Roles	<code>APPDEV</code> , <code>SELECT_CATALOG_ROLE</code>

The following example assumes that you are in the process of creating the user account for Nick or editing the account. Either you have accessed the Create User page and have entered all required fields on the General subpage (see "Example: Creating a User Account" on page 7-13), or you have accessed the Edit User page for Nick (see "Example: Modifying a User Account" on page 7-20). The example also assumes that you have not yet granted any privileges or roles to Nick.

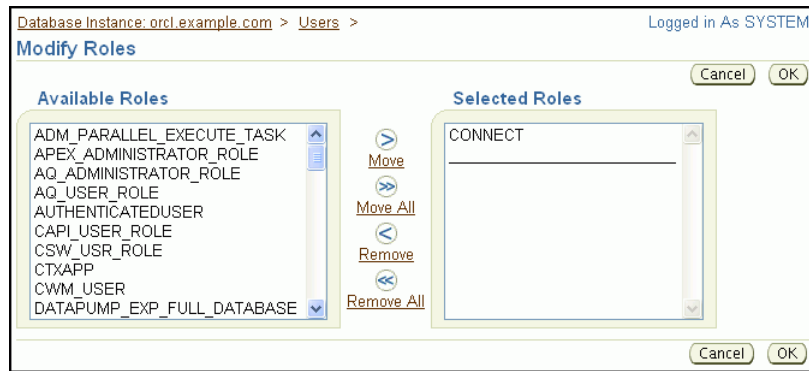
To grant privileges and roles to the user Nick:

1. Toward the top of the Create User or Edit User page, click **Roles** to display the Roles subpage.

The Roles subpage shows that the `CONNECT` role is assigned to Nick. Database Control automatically assigns this role to all users that you create. (The selected Default check box indicates that the `CONNECT` role is a **default role** for Nick, which means that it is automatically enabled whenever Nick logs in.)

2. Click **Edit List**.

The Modify Roles page appears.



3. In the Available Roles list, locate the APPDEV role, and double-click it to add it to the Selected Roles list. Do the same with the SELECT_CATALOG_ROLE role and then click **OK**.

The Create User or Edit User page returns, showing that the CONNECT, APPDEV, and SELECT_CATALOG_ROLE roles are granted to Nick.

Note: Double-clicking a role is a shortcut. You can also select the role and then click the **Move** button. To select multiple privileges, hold down the Shift key while selecting a range of privileges, or press the Ctrl key and select individual privileges.

4. Toward the top of the page, click **System Privileges** to select the System Privileges subpage.
 5. Click **Edit List**.
- The Modify System Privileges page appears.
6. In the Available System Privileges list, scroll to locate the CREATE ANY TABLE and CREATE ANY VIEW privileges, double-click each to add them to the Selected System Privileges list, and then click **OK**.

The Create User or Edit User page returns, showing the newly added system privileges.

Note: To revoke a role, double-click it in the Selected Roles list on the Modify Roles page. To revoke a system privilege, double-click it in the Selected System Privileges list on the Modify System Privileges page.

7. Toward the top of the page, click **Object Privileges** to select the Object Privileges subpage.
8. In the Select Object Type list, select **Table** and then click **Add**.

The Add Table Object Privileges page appears.



9. Click the flashlight icon next to the Select Table Objects list.

The Select Table Objects dialog box appears.

10. In the Schema list, select **HR**, and then click **Go**.

All tables in the hr schema are displayed.



11. Click **Select All**, and then click the **Select** button.

The Select Table Objects dialog box closes, and the names of all tables in the hr schema appear in the Select Table Objects field on the Add Table Object Privileges page.

12. In the Available Privileges list, double-click the **SELECT** privilege to move it to the Selected Privileges list, and then click **OK**.

The Create User or Edit User page returns, showing that the **SELECT** object privilege for all hr tables is granted to user Nick.

Note: To revoke an object privilege, select it on the Create User or Edit User page (Object Privileges subpage), and then click **Delete**.

13. Do one of the following to save the role and privilege grants:
- If you are creating a user account, then click **OK** to save the new user account.
 - If you are modifying a user account, then click **Apply** to save the changes for the user account.

See Also:

- ["About User Privileges and Roles"](#) on page 7-3
- *Oracle Database 2 Day + Security Guide*

Example: Assigning a Tablespace Quota to a User Account

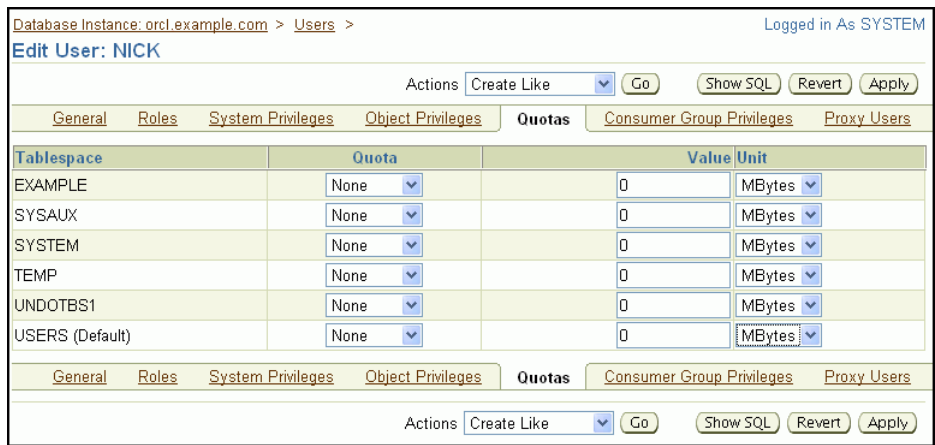
Suppose you are creating or modifying a user account named Nick. You want to assign Nick a space usage quota of 10 MB on his default tablespace.

You must assign Nick a tablespace quota on his default tablespace before he can create objects in that tablespace. (This is also true for any other tablespace in which Nick wants to create objects.) After a quota is assigned to Nick for a particular tablespace, the total space used by all of his objects in that tablespace cannot exceed the quota. You can also assign a quota of UNLIMITED.

The following example assumes that you are in the process of creating the user account for Nick or editing the account. Either you have accessed the Create User page and have entered all required fields on the General subpage (see ["Example: Creating a User Account"](#) on page 7-13), or you have accessed the Edit User page for Nick (see ["Example: Modifying a User Account"](#) on page 7-20). The example also assumes that Nick has not yet been assigned a quota on any tablespaces.

To assign a tablespace quota to user Nick:

1. Toward the top of the Create User or Edit User page, select the **Quotas** subpage.
The Quotas subpage appears, showing that user Nick does not have a quota assigned on any tablespace.



2. In the **Quota** column for tablespace USERS, select **Value** from the list.
3. In the **Value** column for tablespace USERS, enter **10**.

4. Do one of the following to save the new quota assignment:
 - If you are creating a user account, then click **OK** to save the new user account.
 - If you are modifying a user account, then click **Apply** to save changes for the user account.

Example: Modifying a User Account

Suppose you want to remove the quota limitations for the user Nick on his default tablespace, `USERS`. To do so, you must modify his user account.

To modify the user Nick:

1. Go to the Users page, as described in "[Viewing User Accounts](#)" on page 7-12.
2. In the **Select** column, select the user account Nick, and then click **Edit**.

If you do not see user Nick, then he may be on another page. In this case, do one of the following:

- Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the user account for Nick.
- Use the Search area of the page to search for his account. In the **Object Name** field, enter the letters `NI`, and then click **Go**.

You can then select the user account for Nick and click **Edit**.

The Edit User page appears, and displays the general attributes for Nick.

3. Toward the top of the page, select the **Quotas** subpage.
4. In the **Quota** column for tablespace `USERS`, select **Unlimited** from the list, and then click **Apply**.

A message appears, indicating that user Nick was modified successfully.

Locking and Unlocking User Accounts

To temporarily deny access to the database for a particular user account, you can lock the user account. If the user then attempts to connect, then the database displays an error message and does not allow the connection. You can unlock the user account when you want to permit database access again for that user.

To lock or unlock a user account:

1. Go to the Users page, as described in "[Viewing User Accounts](#)" on page 7-12.
2. In the **Select** column, click the desired user account.

If you do not see the desired user account, then it may be on another page. In this case, use the **Next** button to view additional pages or use the Search area of the page to search for the desired user account.

3. Do one of the following:
 - To lock the account, select **Lock User** from the Actions list, and then click **Go**.
 - To unlock the account, select **Unlock User** from the Actions list, and then click **Go**.

A confirmation message appears.

4. Click **Yes**.

Expiring a User Password

When you expire a user password, the user is prompted to change his or her password the next time that user logs in. Reasons to expire a password include the following:

- A user password becomes compromised.
- You have a security policy in place that requires users to change their passwords on a regular basis.

Note: You can automate the automatic expiring of user passwords after a certain interval. See "[Setting the Database Password Policy](#)" on page 7-23.

- A user has forgotten his or her password.

In this third case, you modify the user account, assign a new temporary password, and expire the password. The user then logs in with the temporary password and is prompted to choose a new password.

To expire a user password:

1. Go to the Users page, as described in "[Viewing User Accounts](#)" on page 7-12.
2. In the **Select** column, click the desired user account.

If you do not see the desired user account, then it may be on another page. In this case, do one of the following:

- Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the desired user account.
- Use the Search area of the page to search for the desired user account. In the **Object Name** field, enter the first few letters of the user account name, and then click **Go**.

You can then select the user account.

3. To expire the passwords for all users, select the **Multiple** option, then click **Select All**.
4. Select **Expire Password** from the Actions list, and then click **Go**.
A confirmation message appears.
5. Click **Yes** to complete the task.

Example: Deleting a User Account

Suppose user Nick has moved to another department. Because it is no longer necessary for him to have access to the database, you want to delete his user account.

You must use caution when deciding to deleting a user account, because this action also deletes all schema objects owned by the user. To prevent a user from logging in to the database while keeping the schema objects intact, lock the user account instead. See "[Locking and Unlocking User Accounts](#)" on page 7-21.

To delete user Nick:

1. Go to the Users page, as described in "[Viewing User Accounts](#)" on page 7-12.
2. In the **Select** column, select the user account Nick, and then click **Delete**.

If you do not see the user account Nick, then it may be on another page. In this case, do one of the following:

- Just above the list of user accounts, click **Next** to view the next page. Continue clicking **Next** until you see the user account for Nick.
- Use the Search area of the page to search for the user account. In the **Object Name** field, enter the letters NI, and then click **Go**.

You can then select the user account for Nick and click **Delete**.

A confirmation page appears.

3. Click **Yes** to confirm the deletion of the user account.

Setting the Database Password Policy

This section provides background information and instructions for setting the password policy for all user accounts in the database. It contains the following topics:

- [About Password Policies](#)
- [Modifying the Default Password Policy](#)

See Also:

- ["Administering Database User Accounts"](#) on page 7-11
- *Oracle Database 2 Day + Security Guide*

About Password Policies

When you create a user account, a default password policy is assigned to that user account. The default password policy for a newly installed database includes these directives:

- The password for the user account expires automatically in 180 days.
- The user account is locked 7 days after password expiration.
- The user account is locked for 1 day after 10 failed login attempts.

The default password policy is assigned to user accounts through a database object called a *profile*. Each user account is assigned a profile, and the profile has several attributes that describe a password policy. The database comes with a default profile (named `DEFAULT`), and unless you specify otherwise when you create a user account, the default profile is assigned to the user account.

For better database security, you may want to impose a more strict password policy. For example, you may want passwords to expire every 70 days, and you may want to lock user accounts after three failed login attempts. (A failed login attempt for a user account occurs when a user enters an incorrect password for the account.) You may also want to require that passwords be complex enough to provide reasonable protection against intruders who try to break into the system by guessing passwords. For example, you might specify that passwords must contain at least one number and one punctuation mark.

You change the password policy for every user account in the database by modifying the password-related attributes of the `DEFAULT` profile.

Note: It is possible to have different password policies for different user accounts. You accomplish this by creating multiple profiles, setting password-related attributes differently for each profile, and assigning different profiles to different user accounts. This scenario is not addressed in this section.

Modifying the Default Password Policy

You modify the default password policy for every database user account by modifying the password-related attributes of the profile named `DEFAULT`.

To modify the default password policy:

- Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
- At the top of the page, click **Server** to view the Server subpage.
- In the Security section, click **Profiles**.
The Profiles page appears.
- In the **Select** column, select the profile named **DEFAULT**, and then click **Edit**.
The Edit Profile page appears.
- Toward the top of the page, select the **Password** subpage.

- Change field values as required. Click the flashlight icon next to each field to view a list of choices. (Click **Help** on this page for a description of the fields.)
- Click **Apply** to save your changes.

See Also:

- "[About Password Policies](#)" on page 7-23
- Oracle Database 2 Day + Security Guide*

Users: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Users OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/users/users.htm

Managing Schema Objects

This chapter discusses the creation and management of schema objects. It contains the following sections:

- [About Schema Objects](#)
- [About Schema Object Management Privileges](#)
- [Managing Tables](#)
- [Managing Indexes](#)
- [Managing Views](#)
- [Managing Program Code Stored in the Database](#)
- [Working with Other Schema Objects](#)
- [Schemas: Oracle By Example Series](#)

About Schema Objects

A **schema** is a collection of database objects. A schema is owned by a database user and shares the same name as the user. **Schema objects** are logical structures created by users. Some objects, such as tables or indexes, hold data. Other objects, such as views or synonyms, consist of a definition only.

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.

Naming Schema Objects

Every object in the database belongs to one schema and has a unique name within that schema. Multiple database objects can share the same name, if they are in different schemas. You can use the schema name to unambiguously refer to objects. For example, `hr.employees` refers to the table named `employees` in the `hr` schema. (The `employees` table is owned by `hr`.) The terms *database object* and *schema object* are used interchangeably.

When you create a database object, you must ensure that you create it in the intended schema. One method is to log in to the database as the user who owns the schema and then create the object. Generally, you place all the objects that belong to a single application in the same schema.

A schema object name must abide by certain rules. In addition to being unique within a schema, a schema object name 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 the database raises an error.

The Show SQL Button

You can create and manipulate schema objects with SQL or with Oracle Enterprise Manager Database Control (Database Control).

When creating schema objects using Database Control, you can click the **Show SQL** button to display the SQL statement that is the equivalent of the schema object properties that you specified with the graphical user interface. Database Control submits this SQL statement to create the schema object. This option shows the statement even if it is incomplete, so you must enter all specifications for the schema object to see the complete SQL statement that Database Control submits.

See Also:

- *Oracle Database Concepts* for more detailed information about schema objects, object names, and data types

About Schema Object Management Privileges

As a database administrator (DBA), you can create, modify, and delete schema objects in your own schema and in any other schema. For purposes of this discussion, a database administrator is defined as any user who is granted the DBA role. This includes the *SYS* and *SYSTEM* users by default. Oracle recommends granting the DBA role only to those users who require administrative type access.

You can enable other users to manage schema objects without necessarily granting them DBA privileges. For example, a common scenario is to enable an application developer to create, modify, and delete schema objects in his or her own schema. To do so, you grant the *RESOURCE* role to the application developer.

As described in "[Granting Access to Database Control for Nonadministrative Users](#)" on page 3-8, you must also grant the developer access to Database Control before he or she can log in to Database Control to manage schema objects. If you do not grant access to Database Control, then the developer must manage schema objects with *SQL*Plus* or *SQL Developer*.

See Also:

- "[Example: Granting Privileges and Roles to a User Account](#)" on page 7-16
- "[Administering the Database with SQL-Based Management Tools](#)" on page 3-14

Managing Tables

The following topics discuss database tables and how to create and modify them:

- [About Tables](#)
- [Viewing Tables](#)
- [Viewing Table Data](#)
- [Example: Creating a Table](#)
- [Modifying Table Attributes](#)
- [Example: Loading Data into a Table](#)

- [Deleting a Table](#)

About Tables

The table is the basic unit of data storage in an Oracle database. It holds all user-accessible data. Each table is made up of columns and rows. In the `employees` table, for example, there are columns called `last_name` and `employee_id`. Each row in the table represents a different employee, and contains a value for `last_name` and `employee_id`.

When you create a table, you specify the table type, and define its columns and constraints. **Constraints** are rules that help preserve data integrity.

This section contains the following topics:

- [About Table Types](#)
- [About Table Column Attributes](#)
- [About Table-Level Constraints](#)
- [Other Table Creation Considerations](#)

About Table Types

The most common type of table in an Oracle database is a *relational* table, which is structured with simple columns similar to the `employees` table. Two other table types are supported: *object* tables and *XMLTYPE* tables. Any of the three table types can be defined as *permanent* or *temporary*. Temporary tables hold session-private data that exists only for the duration of a transaction or session. They are useful in applications where a results set must be held temporarily in memory, perhaps because the results set is constructed by running multiple operations.

You can build relational tables in either *heap* or *index-organized* structures. In heap structures, the rows are not stored in any particular order. In index-organized tables, the row order is determined by the values in one or more selected columns. For some applications, index-organized tables provide enhanced performance and more efficient use of disk space.

This section describes permanent, heap-organized tables. For information about other table types and when to use them, see *Oracle Database Administrator's Guide*, *Oracle Database Concepts*, and *Oracle Database Performance Tuning Guide*. For the syntax required to create and alter tables with SQL, see *Oracle Database SQL Language Reference*.

About Table Column Attributes

You define table columns to hold your data. When you create a column, you specify the following attributes:

- [Data Type](#)
- [NOT NULL Column Constraint](#)
- [Default Value](#)
- [Encryption](#)

Data Type The data type attribute defines the kind of data to be stored in the column. When you create a table, you must specify a data type for each of its columns.

Data types define the domain of values that each column can contain. For example, `DATE` columns cannot accept the value `February 29` (except for a leap year) or the

values 2 or SHOE. Each value subsequently inserted in a column assumes the column data type. For example, if you insert 17-JAN-2004 into a date column, then Oracle Database treats that character string as a date value after verifying that it converts to a valid date.

[Table 8-1](#) lists some common Oracle Database built-in data types.

Table 8-1 Common Data Types

Data Type	Description
VARCHAR2 (<i>size</i> [BYTE CHAR])	<p>Variable-length character string having a maximum length of <i>size</i> bytes or characters. A column to hold postal codes for different countries, for example, might be restricted to 12 bytes by defining it as VARCHAR2 (12).</p> <p>You can use the CHAR qualifier, for example VARCHAR2 (10 CHAR), to indicate the maximum length in characters, without regard for the number of bytes required. This is especially useful for languages that use characters with double-byte and triple-byte lengths. The BYTE and CHAR qualifiers override the setting of the NLS_LENGTH_SEMANTICS parameter, which has a default of bytes. The maximum size is 4000 bytes or characters. The minimum is 1 byte or 1 character. You must specify <i>size</i> for VARCHAR2.</p> <p>See <i>Oracle Database Globalization Support Guide</i> for more information.</p>
NUMBER (<i>p</i> , <i>s</i>)	<p>Number having precision <i>p</i> and scale <i>s</i>. Precision sets the maximum number of digits in the number, and scale defines 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 10 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). The precision <i>p</i> can range from 1 to 38. The scale <i>s</i> can range from -84 to 127.</p>
DATE	<p>A composite value that includes both a date and time component. For each DATE value, the database stores the following information: century, year, month, day, hour, minute, and second. When entering a date into a table column of type DATE, you must use the format specified by the NLS_DATE_FORMAT initialization parameter. The NLS_TERRITORY initialization parameter determines the default value of the NLS_DATE_FORMAT parameter. For example, in the United States, the NLS_DATE_FORMAT parameter defaults to 'DD-MON-RR'. You must therefore enter a date in the format '11-JAN-06'. Because this format does not include a time component, the time defaults to 12:00:00 a.m. (midnight). You can also use the TO_DATE function, which converts a character string to a date, to include a time component or to enter a date in another format. The valid date range is from January 1, 4712 BC to December 31, 9999 AD.</p>
CLOB	<p>A character large object (CLOB) containing single-byte or multibyte characters. Both fixed-width and variable-width character sets are supported, both using the database character set. The maximum size is (4 gigabytes - 1) * (database block size). For example, for a block size of 32K, the maximum CLOB size is 128 terabytes.</p>

NOT NULL Column Constraint Constraints determine valid values for the column. In Oracle Enterprise Manager Database Control (Database Control), the only constraint you can define at the column level on the Create Table page is the NOT NULL constraint, which requires that a value be included in the column whenever a row is inserted or updated. Unlike other constraints described in ["About Table-Level Constraints"](#) on page 8-6, which can 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 data must be supplied for a column 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 must be defined with a `NOT NULL` constraint. However, do not define a column as `NOT NULL` if the data can be unknown or may not exist when rows are added or changed. An example of a column for which you must not use a `NOT NULL` constraint is the second, optional line in a mailing address.

The database automatically adds a `NOT NULL` constraint to the column or columns included in the primary key of a 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. You can specify a default value as a literal or as an expression. However, there are limitations on how you construct the expression. See *Oracle Database SQL Language Reference* for details.

Encryption You can enable automatic encryption for column data. See the discussion of transparent data encryption in *Oracle Database 2 Day + Security Guide* for more information.

About Table-Level Constraints

In an Oracle database, you can apply rules to preserve the integrity of your data. For example, in a table that contains 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**, both at the table level and at the column level. Any SQL statement that attempts 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 rows violate the new constraint.

The types of constraints that 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 no two rows can have duplicate values in a specified column or combination of columns. The set of columns is considered to be a 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. It is evaluated each time that a row is inserted or updated. An example of a check constraint is: `SALARY > 0`.
- **Foreign Key**—Requires that for a particular column (or combination of columns), 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 by the foreign key is called the **parent** table. An example of a foreign key constraint is where the department column of the employees table must contain a department ID that exists in the parent department table.

Constraints can be created and usually 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 about constraints

Other Table Creation Considerations

This section describes some additional considerations for creating tables. It contains the following topics:

- [User-Defined Types and Large Objects \(LOBs\)](#)
- [Partitioned Tables and Indexes](#)
- [Physical Storage Attributes](#)
- [Compressed Tables](#)

User-Defined Types and Large Objects (LOBs) Your new table can include one or more columns defined with user-defined types. **User-defined types** enable a single column in a single row to contain multiple values. The multiple values can be represented as arrays, nested tables, or objects, where an object type represents a real-world entity such as a purchase order. (Retrieving a purchase order-type column value could return a *record* that contains purchase order number, customer number, amount, and so on.) User-defined types are created with the `CREATE TYPE` statement and are described in detail in *Oracle Database SQL Language Reference*.

Large object (LOB) columns are used to contain unstructured data (such as text or streaming video), and can hold terabytes of information. In Oracle Database 11g, you can use `SecureFiles`, the next generation LOB data type, which provide high performance, easier manageability, and full backward compatibility with existing LOB interfaces. `SecureFiles` also offer advanced features such as intelligent data compression, deduplication and transparent encryption. The LOB implementation available in Oracle Database 10g Release 2 and prior releases is still supported for backward-compatibility reasons and is now referred to as `BasicFiles`. If you add a LOB column to a table, then you can specify whether it should be created as a `SecureFile` or a `BasicFile`. If you do not specify the storage type, then the LOB is created as a `BasicFile` to ensure backward compatibility.

When creating a table that contains one or more LOB columns, select the **LOB** column, then click **Advanced Attributes** on the General subpage of the Create Table page to specify the storage type (`BasicFile` or `SecureFile`) and the storage options for the LOB column. To specify the same storage type and storage options for all LOB columns in a table, click **Set Default LOB Attributes**.

Partitioned Tables and Indexes You can *partition* tables and indexes. Partitioning helps to support very large tables and indexes by enabling you to divide the tables and indexes into smaller and more manageable pieces called **partitions**. SQL queries and DML statements do not have to be modified to access partitioned tables and indexes. Partitioning is transparent to the application.

After partitions are defined, certain operations become more efficient. For example, for some queries, the database can generate query results by accessing only a subset of partitions, rather than the entire table. This technique (called **partition pruning**) can provide order-of-magnitude gains in improved performance. In addition, data management operations can take place at the partition level, rather than on the entire table. This results in reduced times for operations such as data loads; index creation and rebuilding; and backup and recovery.

Each partition can be stored in its own tablespace, independent of other partitions. Because different tablespaces can be on different disks, this provides a table structure that can be better tuned for availability and performance. Storing partitions in different tablespaces on separate disks can also optimize available storage usage, because frequently accessed data can be placed on high-performance disks, and infrequently retrieved data can be placed on less expensive storage.

Partitioning is useful for many types of applications that manage large volumes of data. Online transaction processing (OLTP) systems often benefit from improvements in manageability and availability, while data warehousing systems benefit from increased performance and manageability.

Physical Storage Attributes You can specify several storage attributes for a table. For example, you can specify the initial size of the table on disk. For more information about setting storage attributes for a table, see *Oracle Database Administrator's Guide* and *Oracle Database SQL Language Reference*.

Compressed Tables Table Compression is suitable for both OLTP applications and data warehousing applications. Compressed tables require less disk storage and result in improved query performance due to reduced I/O and buffer cache requirements. Compression is transparent to applications and incurs minimal overhead during bulk loading or regular DML operations such as INSERT, UPDATE or DELETE. You can configure table compression on the Storage subpage of the Create Table page.

See Also:

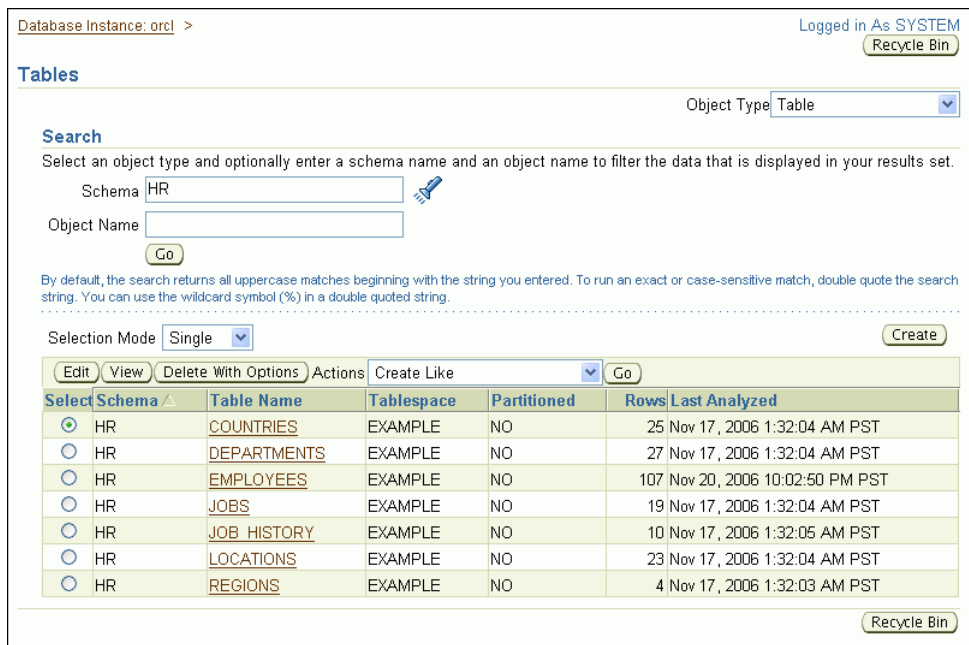
- *Oracle Database Administrator's Guide* for design and management considerations for different table types
- *Oracle Database Concepts* and *Oracle Database VLDB and Partitioning Guide* for more information about partitioned tables and indexes
- *Oracle Database Concepts* for more information about LOBs, SecureFiles and BasicFiles
- ["Example: Creating a Table"](#) on page 8-11

Viewing Tables

You can use Database Control to list all the tables in a specified schema, and to view the definitions of individual tables.

To view tables:

1. Go to the Database Home page, logging in as user `SYSTEM`.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.
The Tables page appears.
4. In the **Schema** field, enter the name of a schema. Alternatively, click the flashlight icon adjacent to the **Schema** field to search for a schema.
Examples of schema names include `SYS` and `hr`.
5. Leave the Object Name field blank to search for and display all tables in the schema. Alternatively, enter a table name or partial table name to limit the search.
If you enter a search string in the **Object Name** field, then all tables that have names that start with the search string are displayed. If you precede the search string with an asterisk (*), then all tables that have the search string anywhere in the table name are displayed.
6. Click **Go**.
The tables in the specified schema are displayed.



- To view the definition of a particular table, select the table and then click **View**. Alternatively, click the table name.

The View Table page appears.

See Also:

- "About Tables" on page 8-3

Viewing Table Data

Besides viewing table names and table definitions, you can view the data stored in the table, and the SQL statement used to display the data. You can also change the SQL statement to alter the results set.

To view table data:

- Search for a table as described in "Viewing Tables" on page 8-8. For example, search for the tables in the hr schema.
- Select the table that contains the data.
For example, select **employees**.
- In the Actions list, select **View Data**, and then click **Go**.

The View Data for Table page appears.

Database Instance: orcl > Tables > Logged in As SYSTEM

View Data for Table: HR.EMPLOYEES Refine Query OK

Query `SELECT "EMPLOYEE_ID", "FIRST_NAME", "LAST_NAME", "EMAIL", "PHONE_NUMBER", "HIRE_DATE", "JOB_ID", "SALARY", "COMMISSION_PCT", "MANAGER_ID", "DEPARTMENT_ID", "HOURLY" FROM "HR"."EMPLOYEES"`

Result

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

The **Query** field displays the SQL query that was run to view the data for the table. The **Result** section shows the data in the table. You may have to use the horizontal scroll bar at the bottom of the page to view all columns.

4. (Optional) Click a column name to sort the data by that column.
5. (Optional) Click **Refine Query** to change the query and redisplay the data.

The Refine Query for Table page appears. This page enables you to select the columns to display. It also enables you to specify a `WHERE` clause for the SQL `SELECT` statement to limit the results.

You can also write and submit your own SQL `SELECT` statement to see the contents of a table. You can run SQL statements by starting a SQL Worksheet session in Database Control. To do so, click **SQL Worksheet** in the Related Links section of the Database Home page.

A detailed description of the `SELECT` statement is in *Oracle Database SQL Language Reference*.

See Also:

- ["About Tables"](#) on page 8-3

Example: Creating a Table

You can use Database Control to create a table. Before you create and populate a table, you can estimate its size to ensure that you have sufficient space to hold its data.

In the following example, you create a table called `purchase_orders` in the `nick` schema that you created in [Chapter 7, "Administering User Accounts and Security"](#). The table has the following columns:

Column Name	Data Type	Size	Not NULL
PO_NUMBER	NUMBER		Yes
PO_DESCRIPTION	VARCHAR2	200	No
PO_DATE	DATE		Yes

Column Name	Data Type	Size	Not NULL
PO_VENDOR	NUMBER		Yes

To create the PURCHASE_ORDERS table in the NICK schema:

- Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.
See "[Accessing the Database Home Page](#)" on page 3-6.
- At the top of the page, click **Schema** to view the Schema subpage.
- In the Database Objects section, click **Tables**.
The Tables page appears.
- Click **Create**.
The Create Table: Table Organization page appears.
- Select **Standard, Heap Organized**, and then click **Continue**.
The Create Table page appears.
- In the **Name** field, enter `purchase_orders` as the table name, and in the **Schema** field, enter `nick`.
Because a default tablespace was specified when creating the user `nick` in the section "[Example: Creating a User Account](#)" on page 7-13, accept the default tablespace setting for the table.
- In the Columns section, enter column information for the `purchase_orders` table as specified in the table in the introduction to this topic. For example, for the first column in the `purchase_orders` table, enter the name `PO_NUMBER` and the data type `NUMBER`, and select the **Not NULL** check box.
For all `purchase_orders` columns, you can leave Scale and Default Value blank.

Database Instance: database > Tables > Create Table Logged in As SYSTEM

Buttons: Show SQL, Cancel, OK

Tabs: General (selected), Constraints, Storage, Options, Partitions

* Name: PURCHASE_ORDERS
 Schema: NICK
 Tablespace: <Default>
 Organization: Standard (Heap Organized)
 Define Using: Column Specification

Buttons: Set Default LOB Attributes, Encryption Options

Select	Name	Data Type	Size	Scale	Not NULL	Default Value	Encrypted
<input checked="" type="radio"/>	PO_NUMBER	NUMBER			<input checked="" type="checkbox"/>		<input type="checkbox"/>
<input type="radio"/>	PO_DESCRIPTION	VARCHAR2	200		<input type="checkbox"/>		<input type="checkbox"/>
<input type="radio"/>	PO_DATE	DATE			<input checked="" type="checkbox"/>		<input type="checkbox"/>
<input type="radio"/>	PO_VENDOR	NUMBER			<input checked="" type="checkbox"/>		<input type="checkbox"/>
<input type="radio"/>		VARCHAR2			<input type="checkbox"/>		<input type="checkbox"/>

Buttons: Add 5 Table Columns

Legend:
 ⚡ Indicates a Primary Key column
 ✓ Indicates a Unique Key column
 🗃 Indicates a Secure File LOB column

Note: To create the table with partitions, click **Partitions** at the top of the page during this step.

8. (Optional) Obtain an estimate of the table size by completing the following steps:

a. Click **Estimate Table Size**.

The Estimate Table Size page appears.

b. In the **Projected Row Count** field, enter **400000** (four hundred thousand), and then click **Estimate Table Size**.

The estimated results are calculated and displayed.

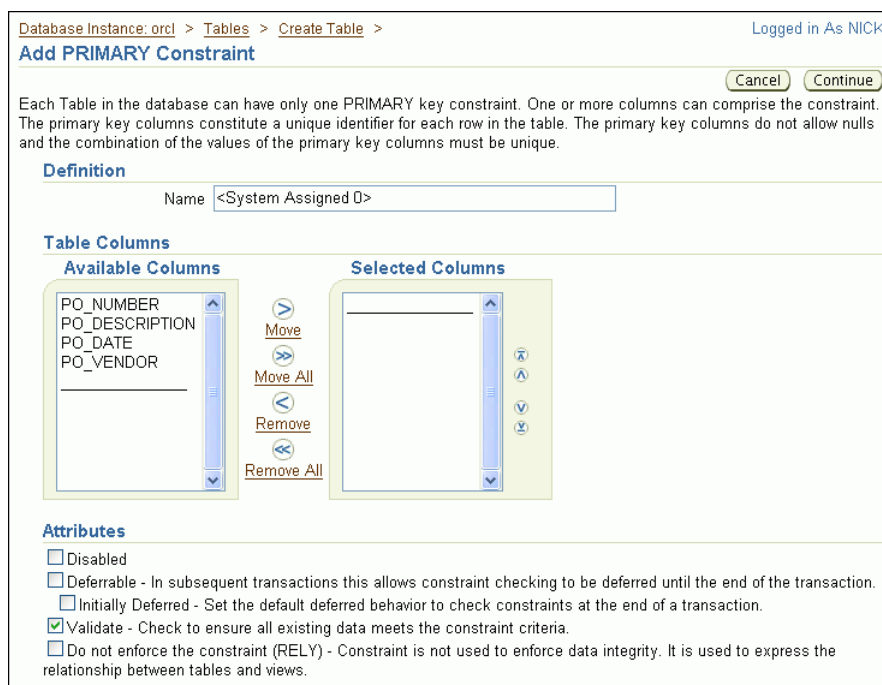
c. Click **OK** to return to the Create Table page.

The estimate of the table size can help you determine what values to use when specifying the storage parameters for the table.

9. Click **Constraints** to view the Constraints subpage, where you can designate a primary key for the table.

10. In the Constraints list, select **PRIMARY**, then click **Add**.

The Add PRIMARY Constraint page appears.



11. In the Available Columns list, select **PO_NUMBER**, and then click **Move**.

Note: You can also double-click **PO_NUMBER**.

The `po_number` column moves to the Selected Columns list.

12. Click **Continue** to return to the Constraints subpage of the Create Table page.

13. Click **OK**.

The Tables page returns, showing a confirmation message and listing the new table in the tables list. The `purchase_orders` table is now created with `po_number` as its primary key.

See Also:

- ["About Tables"](#) on page 8-3

Modifying Table Attributes

You can use Database Control to add and delete table columns and to manage table constraints. This section contains the following topics:

- [Example: Adding Table Columns](#)
- [Example: Deleting a Table Column](#)
- [Example: Adding a New Table Constraint](#)
- [Example: Modifying an Existing Table Constraint](#)
- [Example: Deleting a Table Constraint](#)

See Also:

- ["About Tables"](#) on page 8-3

Example: Adding Table Columns

In this example, you add columns to the `purchase_orders` table that you created previously in ["Example: Creating a Table"](#) on page 8-11. The two new columns are named `po_date_received` and `po_requestor_name`.

To add columns to the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.
The Tables page appears.
4. In the **Schema** field, enter `nick`, and then click **Go**.
All tables owned by user `nick` are displayed.
5. Select the **PURCHASE_ORDERS** table, and then click **Edit**.
The Edit Table page appears.
6. In the Columns section, in the first available row, enter the following information about the new `po_date_received` column:

Field Name	Value
Name	PO_DATE_RECEIVED
Data Type	DATE

You can leave Size, Scale, Not NULL, and Default Value blank.

7. In the next available row, enter the following information about the new `po_requestor_name` column:

Field Name	Value
Name	PO_REQUESTOR_NAME

Field Name	Value
Data Type	VARCHAR2
Size	40

You can leave Scale, Not NULL, and Default Value blank.

8. Click **Apply.**

An update message appears indicating that the table has been modified successfully.

See Also:

- ["About Tables"](#) on page 8-3

Example: Deleting a Table Column

In this example, you delete the `po_requestor_name` column that you added to the `purchase_orders` table in ["Example: Adding Table Columns"](#) on page 8-15.

To delete the `PO_REQUESTOR_NAME` column:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.
The Tables page appears.
4. In the **Schema** field, enter `nick` and then click **Go**.
All tables owned by user `nick` are displayed.
5. Select the **PURCHASE_ORDERS** table, and then click **Edit**.
The Edit Table page appears.
6. In the Columns section, select the **PO_REQUESTOR_NAME** column, and then click **Delete**.
The row that contained the information for the deleted column becomes blank.
7. Click **Apply**.
An update message appears indicating that the table has been modified successfully.

See Also:

- ["About Tables"](#) on page 8-3

Example: Adding a New Table Constraint

In this example, you add a table constraint to the `purchase_orders` table that you created in ["Example: Creating a Table"](#) on page 8-11. To enforce the rule that the `po_date_received` value must be either the same day as, or later than, the value of `po_date`, you add a check constraint.

Note: You can also add constraints during table creation, as shown in "Example: Creating a Table" on page 8-11. In that example, you added a primary key constraint.

To add a table constraint to the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.
See "Accessing the Database Home Page" on page 3-6.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.
The Tables page appears.
4. In the **Schema** field, enter `nick` and then click **Go**.
All tables owned by user `nick` are displayed.
5. Select the **PURCHASE_ORDERS** table, and then click **Edit**.
The Edit Table page appears.
6. Click **Constraints** to display the Constraints subpage.

Select	Name	Type	Table Columns	Disabled	Deferrable	Initially Deferred	Validate	RELY	Check Condition	Referenced Schema	Referenced Table
<input type="radio"/>	SYS_C009266	CHECK	PO_NUMBER	NO	NO	NO	YES	NO	"PO_NUMBER" IS NOT NULL		
<input type="radio"/>	SYS_C009267	CHECK	PO_DATE	NO	NO	NO	YES	NO	"PO_DATE" IS NOT NULL		
<input type="radio"/>	SYS_C009268	CHECK	PO_VENDOR	NO	NO	NO	YES	NO	"PO_VENDOR" IS NOT NULL		
<input checked="" type="radio"/>	SYS_C009269	PRIMARY	PO_NUMBER	NO	NO	NO	YES	NO			

7. In the list adjacent to the **Add** button, select **CHECK**, and then click **Add**.
The Add CHECK Constraint page appears.
8. In the **Name** field, enter `po_check_rcvd_date`, overwriting the system-assigned default name.
9. In the **Check Condition** field, enter the following:
`po_date_received >= po_date`

This expression indicates that `po_date_received` must be greater than or equal to `po_date`. For date columns, this is equivalent to stating that `po_date_received` must be on the same day as, or later than, `po_date`.
10. Click **Continue**.
The new constraint appears on the Constraints subpage.
11. Click **Apply**.
A confirmation message appears.

See Also:

- ["About Tables"](#) on page 8-3
- ["About Table-Level Constraints"](#) on page 8-6

Example: Modifying an Existing Table Constraint

There are a few ways in which you can modify a table constraint. You can change the status of an existing table constraint, for example, from an enabled state to a disabled state. In this example, you disable the check constraint that you created for the `purchase_orders` table in ["Example: Adding a New Table Constraint"](#) on page 8-16.

To disable a constraint for the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**.
The Tables page appears.
4. In the **Schema** field, enter `nick` and then click **Go**.
All tables owned by user `nick` are displayed.
5. Select the `purchase_orders` table, and then click **Edit**.
The Edit Table page appears.
6. Click **Constraints** to display the Constraints subpage.
7. Select the constraint named `PO_CHECK_RCVD_DATE`, and then click **Edit**.
The Edit CHECK Constraint page appears.
8. In the Attributes section, select **Disabled**, and then click **Continue**.
9. Click **Apply**.
A confirmation message appears. The Disabled column shows that the check constraint has been disabled.

See Also:

- ["About Tables"](#) on page 8-3
- ["About Table-Level Constraints"](#) on page 8-6

Example: Deleting a Table Constraint

You can delete constraints from a table with Database Control. Deleting a table constraint may cause the deletion of other constraints. For example, if you delete the primary key constraint from a table (the parent table) that is referenced in a foreign key constraint in another table (the child table), then the foreign key constraint in the child table is also deleted through a cascading delete mechanism.

In this example, you delete the check constraint that you created for the `purchase_orders` table in ["Example: Adding a New Table Constraint"](#) on page 8-16.

To delete a constraint from the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user `nick` or as user `SYSTEM`.
See ["Accessing the Database Home Page"](#) on page 3-6.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Tables**
The Tables page appears.
4. In the **Schema** field, enter `NICK` and then click **Go**.
All tables owned by user `NICK` are displayed.
5. Select the **PURCHASE_ORDERS** table, and then click **Edit**.
The Edit Table page appears.
6. Click **Constraints** to display the Constraints subpage.
7. Select the constraint named **PO_CHECK_RCVD_DATE**, and then click **Delete**.
The check constraint is removed from the list.
8. Click **Apply**.
A confirmation message appears.

See Also:

- *Oracle Database Concepts* for more information about the cascading delete mechanism
- ["About Tables"](#) on page 8-3
- ["About Table-Level Constraints"](#) on page 8-6

Example: Loading Data into a Table

You can use Database Control to load data into a table. You can load data from a source file that is on your local computer—the one where your browser is running—or from a source file that is on the database host computer—the computer on which the Oracle instance is running. Because Database Control invokes the Oracle SQL*Loader utility to load the data, the format of the data in the source file can be of any format that is supported by SQL*Loader. In this example, you use a comma-delimited text file as the source file. In SQL*Loader terminology, the source file is referred to as the *data file*.

SQL*Loader also uses a control file to control the loading of data from the data file. The **control file** is a text file that contains statements written in the SQL*Loader command language. These statements specify where to find the data, how to parse and interpret the data, where to insert the data, and more. Database Control contains a Load Data *wizard* that takes you through the steps of preparing and running a data load job with SQL*Loader. (A **wizard** is an online, guided workflow.) The Load Data wizard can automatically create the SQL*Loader control file for you.

Note: The SQL*Loader control file is unrelated to the database control files described in ["About Control Files"](#) on page 6-3.

In this example, you load data into the `PURCHASE_ORDERS` table that you created in ["Example: Creating a Table"](#) on page 8-11. For simplicity, this example loads only three rows.

To prepare for this example, you must create a text file named `load.dat` on the file system of the database host computer or on the file system of your local computer. The contents of the file should be as follows:

- 1, Office Equipment, 25-MAY-2006, 1201, 13-JUN-2006
- 2, Computer System, 18-JUN-2006, 1201, 27-JUN-2006
- 3, Travel Expense, 26-JUN-2006, 1340, 11-JUL-2006

Note: This example assumes that the columns in the PURCHASE_ORDERS table are the following: PO_NUMBER, PO_DESCRIPTION, PO_DATE, PO_VENDOR, and PO_DATE_RECEIVED. If your PURCHASE_ORDERS table does not have all these columns (or has additional columns), then modify the data in the text file accordingly.

To load data into the PURCHASE_ORDERS table:

1. Go to the Database Home page, logging in as user SYSTEM.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. At the top of the page, click **Data Movement**.
The Data Movement subpage appears.
3. In the Move Row Data section, click **Load Data from User Files**.
The Load Data: Generate or Use Existing Control File page appears.
4. Select **Automatically Generate Control File**, and enter host computer credentials (user name and password) for the database host computer.
5. Click **Continue**.
The first page of the Load Data wizard appears. Its title is Load Data: Data Files.
6. Follow the steps in the wizard, clicking **Next** to proceed to each new step.
For information about using any of the wizard pages, click **Help** on that page. At the conclusion of the wizard, you submit a job that runs SQL*Loader. A job status page is then displayed. If necessary, refresh the status page until you see a succeeded (or failed) status.
7. If the job succeeded, then confirm that the data was successfully loaded by doing one of the following:
 - View the table data.
See "[Viewing Table Data](#)" on page 8-10.
 - Examine the SQL*Loader log file, which is written to the host computer directory that you designated for the SQL*Loader data file.

Note: If the job succeeds, then it means only that Database Control was able to run the SQL*Loader utility. It does not necessarily mean that SQL*Loader ran without errors. For this reason, you must confirm that the data loaded successfully.

8. If the job failed, then examine the SQL*Loader log file, correct any errors, and try again.

See Also:

- *Oracle Database Utilities* for more information about SQL*Loader
- "[About Tables](#)" on page 8-3

Deleting a Table

If you no longer need a table, then you can delete it using Database Control. When you delete a table, the database deletes the data and dependent objects of the table (such as indexes), and removes the table from the data dictionary.

When you delete a table from a locally managed tablespace that is not the `SYSTEM` tablespace, the database does not immediately reclaim the space associated with the table. Instead, it places the table and any dependent objects in the *recycle bin*. You can then restore the table, its data, and its dependent objects from the recycle bin if necessary. You can view the contents of the recycle bin by clicking **Recycle Bin** on the Tables page. Note that users can see only tables that they own in the recycle bin. See *Oracle Database Administrator's Guide* for more information about the recycle bin, including how to view, purge, and recover tables for which you are not the owner.

To delete a table:

1. Search for the table to delete, as explained in "[Viewing Tables](#)" on page 8-8.
2. Select the table, and then click **Delete With Options**.

The Delete With Options page appears.

3. Select **Delete the table definition, all its data, and dependent objects (DROP)**.
4. Select **Delete all referential integrity constraints (CASCADE CONSTRAINTS)**.
5. Click **Yes**.

The Tables page returns and displays a confirmation message.

See Also:

- "[About Tables](#)" on page 8-3

Managing Indexes

The following topics describe how to create and manage indexes:

- [About Indexes](#)
- [Viewing Indexes](#)
- [Example: Creating an Index](#)
- [Example: Deleting an Index](#)

About Indexes

Indexes are optional schema objects that are associated with tables. You create indexes on tables to improve query performance. Just as the index in a guide helps you to quickly locate specific information, an Oracle Database index provides quick access to table data.

You can create as many indexes on a table as you need. You create each index on one or more columns of a table. For example, in a purchase orders table, if you create an index on the vendor number column, then you can then sequentially access the rows of the table in vendor number order, without having to actually sort the rows. Additionally, you can directly access all purchase orders issued to a particular vendor without having to scan the entire table.

After an index is created, it is automatically maintained and used by the database. Changes to the data or structure of a table, such as adding new rows, updating rows,

or deleting rows, are automatically incorporated into all relevant indexes. This is transparent to the user.

Some indexes are created implicitly through constraints that are placed on a table. For example, the database automatically creates an index on the columns of a primary key constraint or unique key constraint.

The following topics provide more background information about indexes:

- [Indexes and Performance](#)
- [Index Attributes](#)

See Also:

- ["Viewing Indexes"](#) on page 8-24
- ["Example: Creating an Index"](#) on page 8-24
- *Oracle Database Concepts*

Indexes and Performance

Indexes generally improve the performance of queries and DML statements that 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 you can improve application performance with more indexes, you can run the SQL Access Advisor in Oracle Enterprise Manager Database Control (Database Control). See ["Running the SQL Access Advisor"](#) on page 10-35.

Before you add additional indexes, examine the performance of your database for queries and DML. You can then compare performance after the new indexes are added.

See Also:

- *Oracle Database Real Application Testing User's Guide* for information about using the SQL Performance Analyzer to analyze the SQL performance impact of any type of schema or system changes

Index Attributes

Indexes can be created in several ways, using various combinations of index attributes. The primary index attributes are the following:

Standard (B-tree) and Bitmap

A standard, B-tree index contains an entry for each value in the index key along with a disk address of 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 index 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 *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 search method 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 column or columns of a table. 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 various combinations of upper and lowercase letters, you can use a function-based index based on the `UPPER()` function 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 the index columns are likely to be included in the `WHERE` clause of frequently executed SQL statements.

Nonpartitioned and Partitioned

As with tables, you can partition an index. In most situations, it is useful to partition an index when the associated table is partitioned, and to partition the index using the same partitioning scheme as the table. (For example, if the table is range-partitioned by sales date, then you create an index on sales date and partition the index using the same ranges as the table partitions.) This is known as a **local** partitioned index. However, you do not have to partition an index using the same partitioning scheme as its table. You can also create a nonpartitioned, or **global**, index on a partitioned table.

See Also:

- *Oracle Database Concepts* for design and management considerations of different index types
- *Oracle Database SQL Language Reference* for the syntax to create indexes
- *Oracle Database VLDB and Partitioning Guide* for more information about partitioned tables and indexes

Viewing Indexes

You use the Indexes page of Database Control to view the indexes in your database.

To view indexes:

1. Go to the Database Home page, logging in as user `SYSTEM`.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Indexes**.
The Indexes page appears.
4. In the Search By list, do one of the following:
 - Select **Index Name** to search for indexes by name.

Every index has a system-assigned or user-assigned name.

- Select **Table Name** to search for indexes that belong to a particular table.
5. In the **Schema** field, enter the name of a schema. Alternatively, click the flashlight icon adjacent to the **Schema** field to search for a schema.
 6. Do one of the following:
 - If you are searching by index name, then leave the Object Name field blank to search for and display all indexes in the schema. Alternatively, enter an index name or partial index name as a search string.
If you enter a search string in the **Object Name** field, then all indexes that have names that start with the search string are displayed. If you precede the search string with an asterisk (*), then all indexes that have the search string anywhere in the index name are displayed.
 - If you are searching by table name, then enter a table name or partial table name in the **Object Name** field.
If you enter a partial table name as a search string, then indexes are displayed for all tables that have names that start with the search string. If you precede the search string with an asterisk (*), then indexes are displayed for all tables that have the search string anywhere in the table name.

7. Click **Go**.

The indexes in the specified schema are displayed.

8. To view the definition of a particular index, select the index and then click **View**. Alternatively, double-click the index name.

The View Index page appears. This page includes basic information about the index, including its status, the table and column or columns on which it is built, the space used by the index, the options used in its definition, and index statistics.

See Also:

- ["About Indexes"](#) on page 8-21

Example: Creating an Index

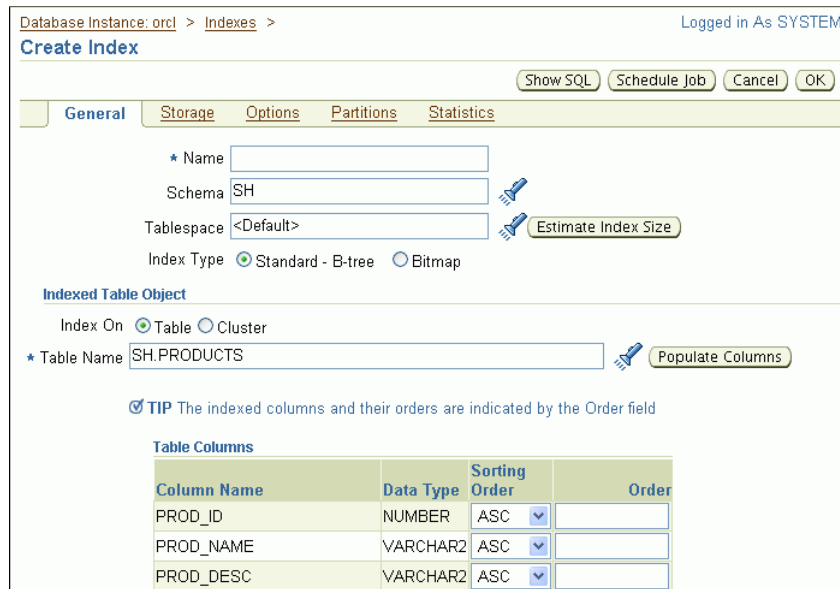
When you create an index, you specify one or more table columns to be indexed and the type of index to create.

In this example, you create a standard B-tree index on the `SUPPLIER_ID` column in the `SH.PRODUCTS` table. (The `SH` schema is part of the sample schemas.)

To create a supplier index on the SH.PRODUCTS table:

1. View the tables in the `SH` schema, by following the instructions in the section ["Viewing Tables"](#) on page 8-8.
2. Select the **PRODUCTS** table.
3. In the Actions list, select **Create Index**, and then click **Go**.

The Create Index page appears.



4. Enter the following information:

- In the **Name** field, enter **PRODUCTS_SUPPLIER_IDX**.
- For the **Tablespace** field, accept the default value.
- For Index Type, select **Standard - B-tree**.
- In the Table Columns list, select the SUPPLIER_ID column by entering **1** in the **Order** column.

If your index were to consist of multiple columns (a concatenated index), then you would enter 2 in the next column to include, and so on. These numbers indicate the order in which the columns are to be concatenated, from left to right, or from most significant in the sort order to least significant.

- For **Sorting Order**, accept the default selection of **ASC** (ascending).

5. Click **OK** to create the index.

The Indexes page returns and displays a confirmation message. The new index is listed in the table of indexes.

See Also:

- ["About Indexes"](#) on page 8-21

Example: Deleting an Index

If you no longer need an index, then you can delete it using Database Control.

In this example, you delete the PRODUCTS_SUPPLIER_IDX index that you created previously on the SH.PRODUCTS table in ["Example: Creating an Index"](#) on page 8-24.

Note: You cannot delete an index that is currently used to enforce a constraint. You must disable or delete the constraint and then, if the index is not deleted as a result of that action, delete the index.

To delete the supplier index on the SH.PRODUCTS table:

1. Go to the Database Home page, logging in as user SYSTEM.

See "[Accessing the Database Home Page](#)" on page 3-6.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Indexes**.
The Indexes page appears.
4. In the Search By list, select **Table Name**.
5. In the **Schema** field, enter **SH**.
6. In the **Object Name** field, enter **PROD**.
You can enter only the first few letters of the table name.
7. Click **Go**.
All indexes on the `PRODUCTS` table are displayed.
8. Select the **PRODUCTS_SUPPLIER_IDX** index, and then click **Delete**.
A confirmation page appears.
9. Click **Yes** to delete the index.
The Indexes page returns and displays a confirmation message.

See Also:

- "[About Indexes](#)" on page 8-21

Managing Views

The following topics describe how to create and manage views:

- [About Views](#)
- [Displaying Views](#)
- [Example: Creating a View](#)
- [Example: Deleting a View](#)

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.

Similar to 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 can provide an additional level of security by restricting access to a predetermined set of rows and columns of a table. They can also hide data complexity and store complex queries.

Many important views are in the `SYS` schema. There are two types: *static data dictionary views* and *dynamic performance views*. Complete descriptions of the views in the `SYS` schema are in *Oracle Database Reference*.

Static Data Dictionary Views

The data dictionary views are called **static views** because they change infrequently, only when a change is made to the data dictionary. Examples of data dictionary changes include creating a new table or granting a privilege to a user.

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 example of a `DBA_` view is `DBA_TABLESPACES`, which contains one row for each tablespace in the database.

- An `ALL_` view displays all the information accessible to the current user, including information from the schema of the current user, and information from objects in other schemas, if the current user has access to those objects through privileges or roles.

An example of an `ALL_` view is `ALL_TABLES`, which contains one row for every table for which the user has object privileges.

- A `USER_` view displays all the information from the schema of the current user. No special privileges are required to query these views.

An example of a `USER_` view is `USER_TABLES`, which contains one row for every table owned by the user.

The columns in the `DBA_`, `ALL_`, and `USER_` views are usually nearly identical.

Dynamic Performance Views

Dynamic performance views monitor ongoing database activity. They are available only to administrators. The names of dynamic performance views start with the characters `V$`. For this reason, these views are often referred to as `V$` views.

An example of a `V$` view is `V$SGA`, which returns the current sizes of various System Global Area (SGA) memory components.

See Also:

- ["Displaying Views"](#) on page 8-28
- ["Example: Creating a View"](#) on page 8-29
- ["Example: Deleting a View"](#) on page 8-30
- *Oracle Database Concepts*

Displaying Views

You can use Oracle Enterprise Manager Database Control (Database Control) to list the views in a specified schema. You can also display the view definitions.

To display views:

1. Go to the Database Home page, logging in as user `SYSTEM`.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Views**.
The Views page appears.
4. In the **Schema** field, enter the name of a schema. Alternatively, click the flashlight icon adjacent to the **Schema** field to search for a schema.

Examples of schema names include `SYS` and `hr`.

5. Leave the Object Name field blank to search for and display all views in the schema. Alternatively, enter a view name or partial view name to limit the search.

If you enter a search string in the **Object Name** field, then all views that have names that start with the search string are displayed. If you precede the search string with an asterisk (*), then all views that have the search string anywhere in the view name are displayed.

6. Click **Go**.

The views in the specified schema are displayed.

7. To view the definition of a particular view, select the view and then click **View**. Alternatively, double-click the view name.

The View page appears.

See Also:

- ["About Views"](#) on page 8-27

Example: Creating a View

In this example, you create a view named `king_view`, which uses the `hr.employees` table as its base table. (The `hr` schema is part of the sample schemas.) This view filters the table data so that only employees who report directly to the manager King, whose employee ID is 100, are returned in queries. 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 the KING_VIEW view on the HR.EMPLOYEES table:

1. Go to the Database Home page, logging in as user `hr` or as user `SYSTEM`.

See ["Accessing the Database Home Page"](#) on page 3-6.

2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Views**.

The Views page appears.

4. Click **Create**.

The Create View page appears.

5. Enter the following information:

- In the **Name** field, enter `king_view`.
- In the **Schema** field, enter `hr`.
- In the **Query Text** field, enter the following SQL statement:

```
SELECT * FROM hr.employees
WHERE manager_id = 100
```

6. Click **OK**.

The Views page returns and displays a confirmation message. The new view appears in the list of views.

To test the new KING_VIEW view:

1. On the Views page, select `king_view` and then select **View Data** from the Actions list.

2. Click **Go**.

The View Data for View page appears. The data selected by the view appears in the Result section.

3. (Optional) You can also test the view by submitting the following SQL statement in SQL*Plus or SQL Developer:

```
SELECT * FROM king_view
```

See Also:

- ["About Views"](#) on page 8-27

Example: Deleting a View

If you no longer need a view, then you can delete it using Database Control.

In this example, you delete the `hr.king_view` view that you created previously in ["Example: Creating a View"](#) on page 8-29.

To delete the HR.KING_VIEW view:

1. Go to the Database Home page, logging in as user `SYSTEM`.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the top of the page, click **Schema** to view the Schema subpage.
3. In the Database Objects section, click **Views**.
The Views page appears.
4. In the **Schema** field, enter `hr`.
5. In the **Object Name** field, enter `king`.
You can enter just the first few letters of the view name.
6. Click **Go**.
`KING_VIEW` is displayed in the list of views.
7. Select `KING_VIEW`, and then click **Delete**.
A Confirmation page appears.
8. Click **Yes** to delete the view.
The Views page returns and displays a confirmation message.

See Also:

- ["About Views"](#) on page 8-27

Managing Program Code Stored in the Database

This section describes your responsibilities as a database administrator (DBA) for program code that is stored in the database. It contains the following topics:

- [About Program Code Stored in the Database](#)
- [Validating \(Compiling\) Invalid Schema Objects](#)

About Program Code Stored in the Database

Oracle Database offers the ability to store program code in the database. Developers write program code in PL/SQL or Java, and store the code in schema objects. You, as the DBA, can use Oracle Enterprise Manager Database Control (Database Control) to manage program code objects such as:

- PL/SQL packages, procedures, functions, and triggers
- Java source code (Java sources) and compiled Java classes

The actions that you can perform include creating, compiling, creating synonyms for, granting privileges on, and showing dependencies for these code objects. You access administration pages for these objects by clicking links in the Programs section of the Schema subpage.

Note that creating and managing program code objects is primarily the responsibility of application developers. However, as a DBA you might have to assist in managing these objects. Your most frequent task for program code objects might be to revalidate (compile) them, because they can become invalidated if the schema objects on which they depend change or are deleted.

Note: Other types of schema objects besides program code objects can become invalid. For example, if you delete a table, then any views that reference that table become invalid.

See Also:

- *Oracle Database Concepts* for an overview of using PL/SQL and Java for server-side programming
- *Oracle Database 2 Day + Java Developer's Guide* for more information about Java sources and Java classes
- *Oracle Database PL/SQL Language Reference* to learn about PL/SQL code
- *Oracle Database Administrator's Guide* for more information about object invalidation

Validating (Compiling) Invalid Schema Objects

As a database administrator (DBA), you may be asked to revalidate schema objects that have become invalid. Schema objects (such as triggers, procedures, or views) might be invalidated when changes are made to objects on which they depend. For example, if a PL/SQL procedure contains a query on a table and you modify table columns that are referenced in the query, then the PL/SQL procedure becomes invalid. You revalidate schema objects by compiling them.

Note: It is not always possible to revalidate a schema object that stores program code by compiling it. You may have to take remedial actions first. For example, if a view becomes invalid because a table that it references is deleted, then compiling the view produces an error message that indicates that the table does not exist. You cannot validate the view until you re-create the table or retrieve it from the recycle bin.

Database Control notifies you when schema objects become invalid by displaying an alert in the Alerts section of the Database Home page.

There are two ways to display schema objects that require validation: by following an alert on the Database Home page, or by viewing the appropriate object page (Views page, Procedures page, and so on) and searching for the objects.

To validate schema objects starting from an alert:

1. Go to the Database Home page, logging in as user SYSTEM.

See "Accessing the Database Home Page" on page 3-6.

2. In the Alerts section, search for alerts with the following message:

n object(s) are invalid in the *schema_name* schema.

An example of such a message is the following:

4 object(s) are invalid in the HR schema.

There is a separate message for each schema that contains invalid objects.

Severity	Category	Name	Impact	Message	Alert Triggered
Warning	Response	User Logon Time (msec)		User logon time is 1234.58 msec.	Nov 17, 2006 2:11:32 AM
Warning	Invalid Objects by Schema	Owner's Invalid Object Count		4 object(s) are invalid in the HR schema.	Nov 23, 2006 2:00:17 AM
Warning	Invalid Objects by Schema	Owner's Invalid Object Count		3 object(s) are invalid in the OLAPSYS schema.	Nov 17, 2006 2:00:17 AM

3. In the **Message** column, click an invalid object message.

The Owner's Invalid Object Count page appears.

4. Under Related Links, click **Invalid Objects Details**.

The Invalid Object Details page appears, showing a list of invalid objects.

Database: orcl > All Metrics > Invalid Objects by Schema > Invalid Objects Count: HR >
Invalid Object Details: HR

Page Refreshed Dec 4, 2006 4:30:29 PM [Refresh](#)

This table shows the current invalid objects in the database. Some objects may become valid after Recompilation.

[Recompile](#) [Delete](#)

Select All | Select None

Select	Name	Type	Status
<input type="checkbox"/>	SECURE_DML	PROCEDURE	INVALID
<input type="checkbox"/>	SECURE_EMPLOYEES	TRIGGER	INVALID
<input type="checkbox"/>	ADD_JOB_HISTORY	PROCEDURE	INVALID
<input type="checkbox"/>	UPDATE_JOB_HISTORY	TRIGGER	INVALID
<input type="checkbox"/>	MyClass	JAVA CLASS	INVALID
<input type="checkbox"/>	IJCLASS	JAVA SOURCE	INVALID

TIP You have selected 0 rows from previously viewed record sets.

5. Select one or more objects, and then click **Recompile**.

A confirmation message appears, and the newly validated objects are removed from the list.

To validate a schema object starting from an object page:

1. Go to the Database Home page, logging in as user SYSTEM.

See "Accessing the Database Home Page" on page 3-6.

2. At the top of the page, click **Schema** to view the Schema subpage.

3. In the Database Objects section, click the link for the object type to validate.
For example, to validate a view, click **Views**.
4. On the object page (for example, the Views page), enter a schema name and, optionally, an object name or partial object name, and then click **Go**.
The schema objects are displayed.
5. Select the schema object to validate.
6. In the Actions list, select **Compile**, and then click **Go**.
A confirmation message appears.

See Also:

- *Oracle Database Concepts* for more information about schema object dependencies
- *Oracle Database Administrator's Guide* for information about managing object dependencies

Working with Other Schema Objects

You can manage other schema objects with Oracle Enterprise Manager Database Control (Database Control), including the following:

- Sequences

A **sequence** is a database object that generates unique integers. Each time that you query the sequence, it increments its current value by a designated amount and returns the resulting integer. Sequences can be simultaneously queried by multiple users, and each user receives a unique value. For this reason, using a sequence to provide the value for a primary key in a table is an easy way to guarantee that the key value is unique, regardless of the number of users inserting data into the table.

- 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. Synonyms can be private or public. A public synonym does not have to be qualified with a schema name, whereas a private synonym does, if the user referencing the private synonym is not the synonym owner. For example, consider the following query, issued by a user who has been granted the `SELECT` object privilege on the `hr.employees` table:

```
SELECT  employee_id, salary
FROM    hr.employees
ORDER BY salary
```

Now suppose you create a public synonym named `personnel` as an alias for the `hr.employees` table, and you grant the `SELECT` privilege on the `hr.employees` table to `PUBLIC` (all database users). With the public synonym in place, any user can issue the following simpler query:

```
SELECT  employee_id, salary
FROM    personnel
ORDER BY salary
```

The user who created this query did not need to know the name of the schema that contains the `personnel` data.

Note: If a user owns a table named `personnel`, then that table is used in the query. If no such table exists, then the database resolves the public synonym and uses the `hr.employees` table.

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 enables application code to run unmodified in both environments. For example, the preceding query would run without errors in a development database that had the `employees` table in the `dev1` schema, if the `personnel` synonym is defined in the development database to point to the `dev1` schema.

Because a synonym is simply an alias, it requires no storage other than its definition in the data dictionary. To reference a synonym in a query, you must have privileges on the object to which it points. Synonyms themselves cannot be secured. If you grant object privileges on a synonym to a user, then you are granting privileges on the object to which the synonym points.

- Database links

A **database link** is a schema object that points to another Oracle database. You use a database link to query or update objects in a remote database. Database links are used in distributed database environments, which are described in *Oracle Database Administrator's Guide*.

See Also:

- *Oracle Database 2 Day Developer's Guide*
- *Oracle Database SQL Language Reference*

Schemas: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this chapter and includes annotated screenshots.

To view the Schemas OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/schema/schema.htm

Performing Backup and Recovery

This chapter introduces you to Oracle Database backup and recovery with Oracle Enterprise Manager Database Control. After reading this chapter, you should be familiar with the basic concepts of Oracle Database backup and recovery operations, know how to implement a disk-based backup strategy, and perform simple repairs to database files.

This chapter contains the following sections:

- [Overview of Database Backup and Recovery](#)
- [Database Backup and Recovery Concepts](#)
- [Configuring Your Database for Basic Backup and Recovery](#)
- [Backing Up Your Database](#)
- [Displaying Backup Reports](#)
- [Managing Backups](#)
- [Performing Oracle Advised Recovery](#)
- [Performing User-Directed Recovery](#)
- [Backup and Recovery: Oracle By Example Series](#)

See Also:

- *Oracle Database Backup and Recovery User's Guide* to learn how to use the command-line interface for Recovery Manager
- *Oracle Database Backup and Recovery Reference* for the syntax and semantics of Recovery Manager commands

Overview of Database Backup and Recovery

The focus in Oracle Database backup and recovery is on the physical backup of database files, which permits you to reconstruct your database. The files protected by the backup and recovery facilities built into Oracle Enterprise Manager Database Control (Database Control) include data files, control files, server parameter files, and archived redo log files. With these files you can reconstruct your database. The backup mechanisms work at the physical level to protect against file damage, such as the accidental deletion of a data file or the failure of a disk drive.

Oracle Recovery Manager (RMAN), a command-line and Enterprise Manager-based tool, is the method preferred by Oracle for efficiently backing up and recovering your Oracle database. RMAN is designed to work intimately with the server, providing block-level corruption detection during backup and restore. RMAN optimizes

performance and space consumption during backup with file multiplexing and backup set compression, and integrates with leading tape and storage media products.

Logical backups, such as exporting database objects such as tables or tablespaces, are a useful supplement to physical backups, but cannot protect your whole database. An effective backup strategy must be based on physical backups.

The Oracle Database 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 data files from backup or performing media recovery.

This section introduces the following flashback features:

- Oracle Flashback Table, which enables you to revert a table to its contents at a time in the recent past
- Oracle Flashback Drop, which enables you to retrieve deleted (dropped) database tables
- Oracle Flashback Database, which enables you to revert the entire database to a past point in time

The first two features operate at the logical level, whereas the last feature operates at the physical level. None of the preceding features requires advance preparation such as creating logical exports to allow for retrieval of your lost data. You can use all of the features while your database is available. *Oracle Database Backup and Recovery User's Guide* discusses the flashback features of Oracle Database at greater length.

Note: Oracle Flashback Database does not recover missing data files.

The Oracle Enterprise Manager physical backup and recovery features are built on the Recovery Manager (RMAN) command-line client. Database Control makes available many of the RMAN features, and provides wizards and automatic strategies to simplify and further automate RMAN-based backup and recovery.

Database Backup and Recovery Concepts

To **back up your database** means to make copies of your data files, control file, and archived redo log files (if your database runs in ARCHIVELOG mode). **Restoring a database** means copying the physical files that comprise the database from a backup medium, typically disk or tape, to their original or to new locations. **Database recovery** is the process of updating database files restored from a backup with the changes made to the database after the backup, typically using online redo log files.

Consistent and Inconsistent Backups

A backup is either consistent or inconsistent. To make a consistent backup, your database must have been shut down cleanly and remain closed for the duration of the backup. All committed changes in the online redo log files are written to the data files during the shut down process, so the data files are in a transaction-consistent state. When you restore your data files from a consistent backup, you can open the database immediately.

If the database is in ARCHIVELOG mode, then you can make inconsistent backups that are recoverable using archived redo log files. Open database backups are inconsistent because the online redo log files contain changes not yet applied to the data files. The

online redo log files must be archived and then backed up with the data files to ensure recoverability.

Despite the name, an inconsistent backup is as robust a form of backup as a consistent backup. The advantage of making inconsistent backups is that you can back up your database while the database is open for updates.

See Also:

- ["About Archived Redo Log Files"](#) on page 6-5
- *Oracle Database Backup and Recovery User's Guide* to learn how to use the `BACKUP` command in the RMAN client

Media Recovery

If you restore the archived redo log files and data files, then you must perform media recovery before you can open the database. Any database transactions in the archived redo log files not reflected in the data files are applied to the data files, bringing them to a transaction-consistent state before the database is opened.

Media recovery requires a control file, data files (typically restored from backup), and online and archived redo log files containing changes since the time the data files were backed up. Media recovery is most often used to recover from 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.

Media recovery can be a complete recovery or a point-in-time recovery. In a complete recovery, you restore backup data files and apply all changes from the archived and online redo log files to the data files. The database is returned to its state at the time of failure and can be opened with no loss of data.

In a point-in-time recovery, you return a database to its contents at a user-selected time in the past. You restore a backup of data files created before the target time and a complete set of archived redo log files from backup creation through the target time. Recovery applies changes between the backup time and the target time to the data files. All changes after the target time are discarded.

Oracle Enterprise Manager Database Control (Database Control) provides an interface to both a complete and a point-in-time recovery in the form of a Recovery wizard. However, this documentation focuses on complete recovery.

See Also:

- ["Performing User-Directed Recovery"](#) on page 9-31
- *Oracle Database Backup and Recovery User's Guide* for more detailed information about complete and point-in-time recovery

Fast Recovery Area

To simplify the management of backup and recovery files, you can create a fast recovery area for your database. The fast recovery area is an Oracle-managed directory, file system, or Oracle Automatic Storage Management disk group that provides a centralized disk location for backup and recovery files. Oracle creates archived logs and flashback logs in the fast recovery area. RMAN can store its backup sets and image copies in the fast recovery area, and it uses it when restoring files during media recovery. The fast recovery area also acts as a disk cache for tape.

Oracle Database automatically manages this storage, deleting files that are no longer needed. Periodically copying backups to tape frees space in the fast recovery area for

other files. Oracle recommends that you enable a recovery area to simplify backup management. Except as noted, this documentation assumes the use of a recovery area.

See Also:

- ["Planning Space Usage and Location for the Fast Recovery Area"](#) on page 9-6
- *Oracle Database Backup and Recovery User's Guide* for more detailed information about configuring a fast recovery area

RMAN Repository

RMAN maintains a record of database files and backups for each database on which it performs operations. This metadata is called the **RMAN repository**.

If you back up a file without using RMAN, such as by copying the file at the host operating system level, then you can add metadata about the copy to the RMAN repository. If you later use a command such as `RESTORE DATABASE`, then RMAN uses the records in the repository to select backups needed for the recovery.

The primary location for the RMAN repository for a database is its control file. The importance of this metadata for RMAN 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 is stored in a schema called the **recovery catalog**. The recovery catalog is located in a separate database and can store metadata for multiple databases. Use of a recovery catalog is optional and is beyond the scope of this documentation.

See Also:

- ["Displaying Backup Reports"](#) on page 9-21
- ["Managing Backups"](#) on page 9-23
- *Oracle Database Backup and Recovery User's Guide* for more detailed information about RMAN reporting capabilities and querying the recovery catalog

Configuring Your Database for Basic Backup and Recovery

This section explains how to set up your database to take advantage of Oracle suggested backup strategies. If you configured the database for automated backups with the Oracle Database Configuration Assistant (DBCA), then skip this section.

To take maximum advantage of Oracle Database features that automatically manage backup and recovery files and operations, configure your database as follows:

- Use a fast recovery area, which automates storage management for most backup-related files, and specify it as an archived redo log file destination.
- Run the database in `ARCHIVELOG` mode so you can perform online backups and have data recovery options such as complete and point-in-time media recovery.

You must also set several policies governing which files are backed up, what format is used to store backups on disk, and when files become eligible for deletion.

This section contains the following topics:

- [Specifying Credentials for Backup and Recovery Using Database Control](#)
- [Planning Space Usage and Location for the Fast Recovery Area](#)

- [Configuring Recovery Settings](#)
- [Monitoring Fast Recovery Area Usage](#)
- [Configuring Backup Settings](#)

Specifying Credentials for Backup and Recovery Using Database Control

You must have the proper credentials to perform some configuration tasks for backup and recovery, and to schedule backup jobs and perform recovery. The following credentials may be required:

- The Oracle Database user you specify when you log in to Oracle Enterprise Manager Database Control (Database Control)
- The host operating system user whose credentials you provide when performing backup and recovery tasks

To enter credentials for backup and recovery tasks:

1. Log in to Database Control as a database user with SYSDBA privileges, or provide host operating system credentials for a user in the dba group on UNIX or Linux, or the ora_dba group on Microsoft Windows.

The host operating system user must also have execute permission for the RMAN command-line client.

Note: The host user may require certain host privileges to run background jobs such as database backups. For example, on UNIX and Linux, the host user must belong to the OSDBA group (typically dba), and on Windows, the host user must be a member of the Administrators group and must be granted the Log on as batch job logon right. See your platform documentation for more information.

For tasks requiring host operating system credentials, a Host Credentials form appears at the bottom of the page used to perform the task. Enterprise Manager uses the credentials when it starts RMAN jobs that you requested or scheduled.

2. Optionally, in the Host Credentials form, select **Save as Preferred Credential**.

If you select this option before performing your action, then the provided credentials are stored persistently for the currently logged-in Oracle Database 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, you may still be prompted for host credentials even if you have saved the preferred credentials.

See Also:

- ["Configuring Your Database for Basic Backup and Recovery"](#)

Planning Space Usage and Location for the Fast Recovery Area

You should place the fast 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 fast recovery area depends on the size and activity levels of your database, which determine the size of your data files and online redo log files in addition to your recovery objectives. Your objectives dictate what kinds of backups you use, when you make them, and how long to keep them.

About the Backup Retention Policy and the Fast Recovery Area

Space management in the fast 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 (period of time). When using a policy based on redundancy, the fast 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. When using a recovery policy based on a period of time (or window), you specify a time interval in days. Files are obsolete only when they are no longer needed for complete recovery or point-in-time recovery to a system change number (SCN) within the window. Therefore, a recovery retention policy based on a window is recommended.

Even after files in the fast recovery area are obsolete, they are typically not deleted until space is needed for new files. If space permits, files recently moved to tape remain on disk to avoid restoring them from tape for a recovery. The automatic deletion of obsolete files and files moved to tape from the fast recovery area makes it a convenient archiving destination. Other destinations require manual deletion of logs.

See Also:

- ["Planning Space Usage and Location for the Fast Recovery Area"](#)

About the Fast Recovery Area Size

Oracle Database Backup and Recovery User's Guide explains how to size the fast recovery area. As a general rule, the larger the fast recovery area, the more useful it is. Ideally, the fast recovery area should be large enough for copies of the data files, control files, online redo log files, and archived redo log files needed to recover the database, and also the copies of these backup files that are kept based on the retention policy.

If your backup strategy includes incremental backups, which are described in ["Incremental Backups of Data Files"](#) on page 9-13, then add enough space to the fast recovery area for these files. If you can move some backups to tape, then you can reduce the size of the fast recovery area. Note that retrieving files from tape causes longer database restore operations and recovery times.

See Also:

- ["Planning Space Usage and Location for the Fast Recovery Area"](#)

Configuring Recovery Settings

In the Recovery Settings page, you can configure settings for instance recovery, media recovery, and fast recovery. In this section, you configure the fast recovery area and enable archiving for the database.

You can configure a fast recovery area when first creating the database. If you did not perform this task at database creation time, however, then you can create a fast recovery area for your database now.

To configure a fast recovery area and put the database in ARCHIVELOG mode:

1. On the host operating system, create a directory to hold the fast recovery area. If you are using Oracle ASM for your storage, then create a new disk group.

Make sure that the operating system permissions for the directory or disk devices allow the database to create files.

2. Log in to Enterprise Manager Database Control as user SYS.
3. On the Database Home page, click **Availability** to display the Availability subpage.
4. In the Backup/Recovery section, click **Recovery Settings**.

The Recovery Settings page appears.

5. Complete the following steps:

- a. In the Media Recovery section, select **ARCHIVELOG Mode**.

- b. If `USE_DB_RECOVERY_FILE_DEST` is not set as an archiving destination, then set it.

This initialization parameter indicates that the fast recovery area should be an archiving destination.

For ease of database management, the best practice is to use the fast recovery area as your *only* archiving destination.

- c. In the Fast Recovery section, enter the path to the fast recovery area created in Step 1 in **Fast Recovery Area Location**, and select a value for **Fast Recovery Area Size**.
- d. Select **Enable Flashback Database**.

This option specifies that the database should generate flashback logs in the fast recovery area, enabling you to use Flashback Database. During usual operation, the database occasionally logs images of data blocks to the flashback logs. The database automatically creates, deletes, and resizes flashback logs.

- e. Ensure that the **Apply initialization parameter changes to SPFILE only** box is not selected.
- f. Click **Apply** to save your changes.

A message prompts you to restart the database.

6. Click **Yes**.

The Restart Database: Specify Host and Target Database Credentials page appears.

7. Enter your host and database credentials, and then click **OK**.

See "[Specifying Credentials for Backup and Recovery Using Database Control](#)" on page 9-5 for information about credentials.

8. On the Restart Database: Confirmation page, click **Yes** to begin the restart operations.

You can click **Refresh** periodically to monitor the progress of the operation.

9. Make a consistent (that is, offline) backup of your entire database *immediately* after switching your database to ARCHIVELOG mode.

Caution: You cannot use backups from *before* the switch to ARCHIVELOG mode to restore and recover the database to a point in time *after* the switch. Thus, if you do not *immediately* make a backup after switching, then you are running your database without a valid backup. See "[Performing and Scheduling Backups Using Database Control](#)" on page 9-14 to learn how to make database backups.

See Also:

- "[Rewinding a Database Using Oracle Flashback Database](#)" on page 9-37
- *Oracle Database Backup and Recovery User's Guide* for more details about configuring a fast recovery area

Monitoring Fast Recovery Area Usage

It is important to monitor space usage in the fast recovery area to ensure that it is large enough to contain backups and other recovery-related files.

To monitor available space in the fast recovery area:

In the High Availability section of the Database Home page, click the link next to **Usable Fast Recovery Area (%)**.

The Recovery Settings page appears.

The Reclaimable Recovery Area (GB) and Free Recovery Area (GB) settings indicate how much space is available.

See Also:

- "[Configuring Your Database for Basic Backup and Recovery](#)"

Configuring Backup Settings

You can configure several backup-related settings and policies. For example, you can determine how backups are stored, which data is backed up, and how long backups are retained. You can also configure settings to optimize backup performance.

This section explains the concepts behind the available settings and also explains how to change them using the Backup Settings page in Database Control. The settings on the Device subpage affect how RMAN writes backups to disk and to tape.

About RMAN Backups

Database backups created by RMAN are stored as image copies or backup sets.

Image copies are exact byte-for-byte copies of files. You can create an image copy by copying a file at the operating system level. Unlike copying files at the operating system level, however, image copies created through RMAN or Database Control are recorded in the RMAN repository so that RMAN can use these copies during database restore operations and recovery. RMAN can restore files only if they are recorded in the RMAN repository. RMAN can create image copies only on disk.

Backup sets are logical entities produced by the RMAN `BACKUP` command. This command can produce one or more backup sets on disk or media management devices. RMAN can write backup sets only to a media manager.

Each backup set contains several physical files called **backup pieces**. A backup piece stores the backup of one or more database files in a compact RMAN-specific format. One advantage of backup sets is that RMAN uses unused block compression to save space in backing up data files. Only those blocks in the data files that have been used to store data are included in the backup set.

RMAN depends on server sessions, processes that run on the database server, to create backups and restore them. Each server session, in turn, corresponds to an RMAN channel, representing one stream of data to or from a backup device. Channels are either of type disk or type SBT (tape).

RMAN supports parallelism, which is the use of multiple channels and server sessions to perform the work of one backup or recovery task. Proper use of parallelism can greatly increase performance on backup and recovery tasks.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for an overview of RMAN architecture
- *Oracle Database Backup and Recovery User's Guide* for more conceptual details about RMAN backups

Configuring Backup Device Settings

For disk-based backups, you can configure the default format for backups, the location on disk where backups are stored, and whether backup tasks run in parallel.

For tape backups, you can configure settings such as the number of tape drives and whether backups are compressed. On most platforms, you must integrate a media manager with the Oracle database to use sequential media for storage.

You can use Oracle Secure Backup, which supports both database and file system backups to tape, as your media manager. Oracle Secure Backup provides the same services for RMAN as other third-party SBT interfaces, but is better integrated with Database Control. This section assumes that you make only disk backups.

To configure backup settings for disk:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Backup/Recovery section, click **Backup Settings**.
The Backup Settings page appears.
3. At the top of the page, click **Device**.
The Device subpage of Backup Settings appears.
4. Complete the following steps:
 - a. In **Parallelism**, enter **1**.

Later, when you have had time to review the information in *Oracle Database Backup and Recovery User's Guide* about parallelism and performance in RMAN, you may want to change this value.

- b. Leave the Disk Backup Location field blank so that RMAN directs backups to the fast recovery area.

- c. In the **Disk Backup Type** section, select **Backup Set**.

One advantage of backup sets is that RMAN uses unused block compression to save space when backing up data files. Only those blocks in the data files that have been used to store data are included in the backup set.

5. In Host Credentials section at the bottom of the page, enter values in the **Username** and **Password** fields.

Host credentials are described in "[Specifying Credentials for Backup and Recovery Using Database Control](#)" on page 9-5.

6. Click **Test Disk Backup** (near the top of the page) to ensure the credentials and backup location are correct.

A message appears that states whether the test was successful.

In this example you do not change the settings on the Backup Set subpages.

See Also:

- *Oracle Database Backup and Recovery User's Guide* to learn how to use the RMAN client to configure the default device for backups

Configuring Backup Policy Settings

You can set the backup policies that govern control file and server parameter file backups, tablespaces to exclude from an entire database backup, and the backup retention policy.

To configure the backup policy settings:

1. On the Database Home page, click **Availability** to display the Availability subpage.

2. In the Backup/Recovery section, click **Backup Settings**.

The Backup Settings page appears.

3. At the top of the page, click **Policy**.

The Backup Policy subpage appears.

4. Perform the following actions:

- Select the **Automatically backup the control file and server parameter file (SPFILE) with every backup and database structural change** option. Leave the Autobackup Disk Location field blank so that RMAN stores the automatic backups in the fast recovery area.

The server parameter file and control file are critical to the database and RMAN, and are also relatively small compared to typical data files. Backing them up frequently results in relatively little storage overhead.

- Select the **Optimize the whole database backup by skipping unchanged files such as read-only and offline data files that have been backed up** option.

This option saves space in the fast recovery area.

- Select the **Enable block change tracking for faster incremental backups** option. Either leave the Block Change Tracking File field blank (if you specified Oracle Managed Files when creating the database) or enter a file name (including path).

This option takes advantage of the block change tracking feature, which greatly improves incremental backup performance at a small cost of overhead.

5. In the Tablespaces Excluded From Whole Database Backup section, leave the settings as they are.

This feature enables you to specify a list of tablespaces to exclude from a backup. For example, you do not have to include read-only tablespaces in every backup.

6. In the Retention Policy section, select **Retain backups that are necessary for a recovery to any time within the specified number of days (point-in-time recovery)**. In **Days**, enter **31**.

This setting enables a recovery policy with retention based on a period of time.

7. In the Archivelog Deletion Policy section, select **None**.

This option specifies that logs are eligible for automatic deletion only when they have been backed up to tape or are obsolete based on the retention policy.

8. At the bottom of the page, click **OK** to save your changes.

See Also:

- ["Managing Backups"](#) on page 9-23
- *Oracle Database Backup and Recovery User's Guide* to learn how to configure a backup retention policy

Backing Up Your Database

This section describes how to back up your database with Oracle Enterprise Manager Database Control (Database Control). The Oracle suggested strategy for disk-only backups provides efficient daily backup of the database. This strategy enables you to quickly return your database to its state at any point during the previous 24 hours. For more flexible backup options, see *Oracle Database Backup and Recovery User's Guide*.

RMAN Backup Concepts

This section provides the concepts you need to understand the Oracle suggested backup strategy and other backup types provided through Database Control.

Full Backups of Data Files

A full backup of a data file includes all used blocks of the data file. As explained in ["About RMAN Backups"](#) on page 9-9, an RMAN backup can be either an image copy or backup set. An image copy is a bit-for-bit copy of the data file, and thus includes unused blocks.

Incremental Backups of Data Files

RMAN incremental backups copy only those blocks in a data file that change between backups. A level 0 incremental backup, which copies all blocks in the data file, is used as a starting point for an incremental backup strategy.

Incremental backups at level 1 copy only images of blocks that have changed since a previous incremental backup. 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 blocks changed since the most recent level 0 or level 1 incremental backup are included. A typical incremental strategy makes level 1 backups at regular intervals such as once each day.

Recovering changed blocks from incremental backups can speed media recovery. Because an incremental level 1 backup copies the final contents of all data file blocks

changed during the period covered by the incremental backup strategy, the recovery process can avoid reapplying updates from the archived redo log files of that period, and update each block with its final contents. The archived redo log files are used only for changes from the period not covered by level 1 backups.

Incrementally Updated Backups: Rolling Forward Image Copies of Data Files

RMAN enables you to apply level 1 incremental backups to an older image copy of your data files. You can roll forward the copy to the point in time of the most recent 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 forward the file in time, so that its contents are equivalent, for the purposes of database recovery, to an image copy of the data file made at the time of the last incremental level 1 backup.

Incorporating incrementally updated backups into your backup strategy shortens expected recovery times. Media recovery to the present time or to a point in time in the recent past can begin at the time of the last level 1 backup applied, rather than at the time of the last full database backup.

Backup Tags

A **tag** is a text string that identifies that backup, either uniquely or as part of a group of backups. All RMAN backups, including incremental backups, are labeled with a tag. For example, if you performed a full database backup every Saturday, then you could use the tag `FULL_SAT` to identify this backup.

You can use tags to refer to specific backups in RMAN commands. For example, you could issue a command to move the latest `FULL_SAT` backup to tape. If you do not specify a tag, then RMAN creates a unique tag automatically.

Because you can use tags to refer to different groups of backups, you can create different routines in your 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 the tag.

Performing and Scheduling Backups Using Database Control

Database Control enables you to perform RMAN backups and schedule backup jobs required by your backup strategy.

Performing a Whole Database Backup

Whole backups of a database include the complete contents of all data files of the database, also the control file, archived redo log files, and server parameter file. With these files, you can perform a complete recovery.

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 enable or disable `ARCHIVELOG` mode (see "[Configuring Recovery Settings](#)" on page 9-7). This section explains how to make whole database backups to disk.

To perform a whole database backup when the database is open:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section of the Availability page, click **Schedule Backup**.

The Schedule Backup page appears.

Schedule Backup

Oracle provides an automated backup strategy based on your disk and/or tape configuration. Alternatively, you can implement your own customized backup strategy.

Oracle-Suggested Backup

Schedule a backup using Oracle's automated backup strategy.

[Schedule Oracle-Suggested Backup](#)

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.

[Schedule Customized Backup](#)

- Whole Database
- Tablespaces
- Datafiles
- Archived Logs
- All Recovery Files on Disk
Includes all archived logs 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
- Sets up recovery window for backup management
- Schedules recurring and immediate backups
- Automates backup management

Customized:

- Specify the objects to be backed up
- Choose disk or tape backup destination
- Override the default backup settings
- Schedule the backup

3. In the Host Credentials section at the bottom of the page (not shown in the figure), enter values in the **Username** and **Password** fields if the Preferred Credentials are not set.
4. In the Customized Backup section, select **Whole Database**, and then click **Schedule Customized Backup**.
The Schedule Customized Backup: Options page appears.
5. In the Backup Type section, select **Full Backup**.
6. In the Backup Mode section, select either **Online Backup** or **Offline Backup**.
Typically, you want to perform online backups to maximize database availability.

Note: As explained in "[Configuring Your Database for Basic Backup and Recovery](#)" on page 9-5, you can make online backups only if your database is running in ARCHIVELOG mode.

7. In the Advanced section, make the following selections, and then click **Next**:
 - If you are performing an online backup, then select **Also back up all archived logs on disk**.
You do not have 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 the database from this backup. However, you can choose to include archived logs in the backup.
 - If a fast recovery area is your only archiving destination, then do not select "Delete all archived logs from disk after they are successfully backed up".
In this case, archived redo log files that have been backed up are deleted automatically as space is needed. If you are using some other archiving destination, then it may be useful to select this option as part of managing your backup storage.
 - If the fast recovery area stores backups, then do not select the "Delete obsolete backups" option.

In this case, obsolete backups are deleted automatically as space is needed. If you use another destination for your backup files, then you can select this option.

- Do not select "Use proxy copy supported by media management software to perform a backup" for now.
- Do not enter a value in the Maximum Files per Backup Set field or the Section Size field.

The Schedule Customized Backup: Settings page appears.

8. Select **Disk and then click **Next**.**

Oracle recommends backing up to disk when possible to minimize recovery time by minimizing restore operations from tape. Later, you can move disk backups to tape.

The Schedule Customized Backup: Schedule page appears.

9. Complete the following steps:

- a. In the Job section, leave the default values for **Job Name** and **Job Description**.**

In the **Job Name** field, you can enter a user-specified tag for this backup. The job name is used as a prefix for the backup tag for backups created by this job.

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`.

In the **Job Description** field, you can enter any descriptive text that is useful for your own reference.

- b. In the Schedule section, select **One Time (Immediately)**, and then click **Next**.**

The Schedule Customized Backup: Review page appears.

10. Click **Submit Job.**

The Status page appears. This page should contain a message that indicates the job was successfully submitted.

11. Click **View Job to monitor the progress of the backup.**

The Execution page for your database appears. This page contains a Summary section describing the job. The table at the bottom of the page displays 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, you can see the current phase of the RMAN job. Clicking the name of the phase of the backup job displays a page containing the RMAN output for that part of the job. If you clicked on a job phase link, then from your current page, click the Execution: *Database_name* link at the top of the page to return to Execution page for your database.

To perform a whole database backup when the database is closed:

- Follow the same steps as for an open database backup, except select **Offline Backup** in Step 5 and then continue with the steps that follow.

Using the Oracle Suggested Backup Strategy

Database Control 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 user-specified recovery window (time period). The Oracle suggested strategy leverages the 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 archive log to your data files.

About the Oracle Suggested Backup Strategy The Oracle suggested strategy is based on creating an image copy of your database. The database rolls forward this copy by using incrementally updated backups. Oracle Enterprise Manager schedules RMAN backup jobs during the overnight hours.

For each data file, the strategy calls for backups as follows:

- At the beginning of day 1 of the strategy (the time the first scheduled job actually runs), RMAN creates an incremental level 0 data file copy. It contains the data file contents at the beginning of day 1.
If a recovery is required, then the archived redo log files from day 1 can be used to recover to any point during day 1.
- At the beginning of day 2, RMAN creates a differential incremental level 1 backup that contains the blocks changed during day 1.
If a recovery is required, then RMAN can apply this incremental level 1 to roll forward the level 0 backup to the beginning of day 2. RMAN can use archived redo log files to recover to any point during day 2.
- At the beginning of each day n for day 3 and onward, RMAN applies the level 1 backup from the beginning of day $n-1$ to the level 0 backup. This action brings the data file copy to its state at the beginning of day $n-1$. Then, RMAN creates a new level 1 backup that contains the blocks changed during day $n-1$.
If a recovery is required, then RMAN can apply this incremental level 1 backup to the data file rolled forward on day $n-1$ to the beginning of day n . RMAN can use archived redo log files to recover the database to any point during day n .

In the Oracle suggested backup strategy, data file copies have the tag `ORA$OEM_LEVEL_0`. The differential level 1 incremental backups in this strategy share the same tag. You can safely implement other backup strategies without interfering with the Oracle suggested backup strategy.

Oracle suggested backup strategies also use tape backups in addition to disk backups, but these are beyond the scope of this section.

Backing Up Your Database Using the Oracle Suggested Disk Backup Strategy The following procedure schedules daily database backups. The strategy uses incremental backups and incrementally applied backups to enable quick recovery to any time in the preceding 24 hours.

To backup your database using the Oracle-suggested disk backup strategy:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section of the Availability page, click **Schedule Backup**.
The Schedule Backup page appears.
3. In the Oracle-Suggested Backup section, click **Schedule Oracle-Suggested Backup**.

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.

4. Select **Disk** and then click **Next**.

The Schedule Oracle-Suggested Backup: Setup page appears.

This page describes the backups that are performed each day as part of the disk-based strategy.

5. Review the information and then click **Next**.

The Schedule Oracle-Suggested Backup: Schedule page appears.

6. Choose a **Start Date**, **Time Zone**, and **Daily Backup Time** for the daily backups, and then click **Next**.

The Schedule Oracle-Suggested Backup: Review page appears.

Database Control displays the backup script that RMAN will run (although you cannot edit the script directly). The script should look similar to the following:

```
Daily Script:
run {
  allocate channel oem_disk_backup device type disk;
  recover copy of database with tag 'ORA_OEM_LEVEL_0';
  backup incremental level 1 copies=1 for recover of copy with tag
  'ORA_OEM_LEVEL_0' database;
}
```

7. Click **Submit Job**.

Scheduling Miscellaneous Backup Tasks

After taking some time to understand the full range of available backup options described in *Oracle Database Backup and Recovery User's Guide* you may decide to schedule backup tasks beyond those used to implement the Oracle suggested backup strategy.

To schedule miscellaneous backup tasks:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section of the Availability page, click **Schedule Backup**.
The Schedule Backup page appears.
3. In the Customized Backup section, select the object to back up. For example, select **Tablespaces** or **Archivelogs**.
4. Make sure the host credentials have been filled in, then click **Schedule Customized Backup** to start the Schedule Customized Backup wizard.
5. On the following pages of the wizard, specify details such as objects to back up, and required options and settings. Click **Next** on each page to proceed to the next page.

The choices presented on these pages are determined by the type of objects being backed up. For example, to backup a tablespace, on the first page of the wizard you would click the **Add** button to specify the tablespaces you want to include in the backup job. On the following pages, you would select the backup options, location, and schedule.

For more information about using the wizard, click **Help** on any page.

6. Click **Submit Job**.

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. You can validate your backups in the following ways:

- Select specific backup sets or image copies in Database Control and validate them. This technique indicates if a backup exists and can be restored. For this form of validation, use the Manage Current Backups pages as described in "[Validating Selected Backups](#)" on page 9-20.
- Specify database files and let RMAN select backups to use when restoring those files, as it would for an actual restore operation. This technique ensures that your available backups are sufficient to restore the database. For this form of validation, use the Perform Recovery pages as described in "[Validating Backups for Restore Operations](#)" on page 9-20.

Note: Validating backups stored on tape can be time-consuming because the entire backup is read from tape.

You can set up both forms of validation as scheduled tasks in Database Control. You should incorporate both forms of validation into your backup strategy to ensure that your recoverability goals are met by your available backups.

Validating Selected Backups

Validating specific backups checks whether these backups exist and can be restored. It does not test whether the set of available backups meet your recoverability goals. For example, image copies of data files for several tablespaces from your database may exist, each of which can be validated. If there are some tablespaces for which no valid backups exist, however, then you cannot restore and recover the database.

To validate selected backups:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Availability page, in the Manage section, click **Manage Current Backups**.
The Manage Current Backups page appears.
3. Use the search features on this page to identify the backups whose contents you want to validate.
4. Select the check boxes next to each backup of interest in the list of current backups, and then click **Validate**.
The Validate: Specify Job Parameters page appears.
5. Specify a job name and description (or use the default values), and time settings for starting and repeating the operation. Click **Submit Job**.
A message confirms the submission of the job.
6. Click the **View Job** link to display details about the validation.

See Also:

- *Oracle Database Backup and Recovery User's Guide* to learn how to use the `VALIDATE` command

Validating Backups for Restore Operations

You can test whether a sufficient set of backups exists that can be used to restore the specified database files. After you specify which tablespaces to restore and, possibly, a time as of which to restore them, RMAN selects a set of backups that contain the needed data. RMAN reads the selected backups in their entirety to confirm that they are not corrupt, but does not produce output files.

Validating the restoration 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.

To verify whether specified database files can be restored:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section of Availability page, click **Perform Recovery**.
The Perform Recovery page appears.
3. In the User-Directed Recovery section, select **Datafiles** or **Tablespaces** from the Recovery Scope list, depending on what you want to validate.
4. For the Operation Type, select **Restore Datafiles** or **Restore Tablespaces** and then click **Recover**.
The Perform Object Level Recovery page appears.
5. Click **Add** to add tablespaces or data files for the validate operation. Select the data files or tablespaces to restore, then click **Select**. After making your selections, click **Next**.
The Perform Object Level Recovery: Restore page appears.
6. In the Backup Selection section, specify which backups should be restored. Typically you would select the most recent backup.
7. In the Backup Validation section, select **Validate the specified backup without restoring the data files**. Then, click **Next**.
The Perform Object Level Recovery: Schedule page appears.
8. The job to validate the backup runs immediately. You can specify a Job Name and Job Description, or use the default values. Click **Next**.
The Perform Object Level Recovery: Review page appears.
9. Click **Submit Job** to run the validation.
The Perform Recovery: Result page appears.
10. (Optional) Click **View Job** to see the results of the Backup Validation job.

See Also:

- *Oracle Database Concepts* for datafile recovery concepts
- *Oracle Database Backup and Recovery User's Guide* to learn how to use the `RESTORE . . . VALIDATE` command

Displaying Backup Reports

Backup reports contain summary and detailed information about past backup jobs run by RMAN, including both backups run through Oracle Enterprise Manager Database Control (Database Control) and the RMAN command-line client.

To display backup reports:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Availability page, in the Manage section, click **Backup Reports**.

View Backup Report

The following backup jobs are known to the database. The data is retrieved from the database control file.

Search

Status: Start Time: Type:

Results

Total 3 (Completed ✓ 3)

Backup Name	Status	Start Time	Time Taken	Type	Output Devices	Input Size	Output Size	Output Rate (Per Sec)
BACKUP_ORCL_EXAMPL_050109083242	COMPLETED	May 1, 2009 8:33:05 AM PDT	00:00:06	DATAFILE INCR	DISK	14.28M	12.70M	2.12M
BACKUP_ORCL_EXAMPL_050109081759	COMPLETED	May 1, 2009 8:18:22 AM PDT	00:05:00	DB FULL	DISK	1.57G	1.29G	4.41M
2009-05-01T07:47:28	COMPLETED	May 1, 2009 7:47:32 AM PDT	00:00:05	CONTROLFILE	DISK	9.28M	9.33M	1.87M

[TIP](#) * in Output Devices column indicates that backups from this job are on DISK and SBT_TAPE

The Backup Reports page contains a list of recent backup jobs. You can use the Search section of the page to restrict (filter) the backups listed by the status of the job, the start time of the backup, and the type. In the Search sections, specify any filter conditions and click **Go**.

3. Click a link in the **Backup Name** column to display detailed information about any backup.

The Backup Report page is displayed for the selected backup. Use the Back button of your browser to return to the Backup Reports page.

4. Click the link in the **Status** column to view a log of RMAN output for the job.

Note: The control file view `V$RMAN_OUTPUT` contains the output of recent RMAN jobs. The contents of this view are not preserved when the instance is restarted. Therefore, the output from past RMAN jobs may not be available.

Managing Backups

As part of a backup strategy, you must manage database backups. A related task is managing the record of those backups in the RMAN repository. Oracle Enterprise Manager Database Control (Database Control) simplifies these tasks.

About Backup Management

An essential part of a backup and recovery strategy is managing backups after you create them. Backup management includes deleting obsolete backups and performing periodic checks to ensure that backups are available and usable.

You can perform backup management tasks from the Manage Current Backups page.

To display the Manage Current Backups page:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Availability page, in the Manage section, click **Manage Current Backups**.

This page has two subpages: Backup Sets and Image Copies. Both pages have similar purposes: listing backups based on the record in the RMAN repository and enabling you to manage the backups.

Manage Current Backups

Catalog Additional Files Crosscheck All Delete All Obsolete Delete All Expired

This backup data was retrieved from the database control file.

Backup Sets Image Copies

Search

Status: Available

Contents: Datafile Archived Redo Log SPFILE Control File

Completion Time: Within a month **Go**

Results

Crosscheck Change to Unavailable Delete Validate

Select All | Select None

Select	Key Tag	Completion Time	Contents	Device Type	Status	Keep	Pieces
<input type="checkbox"/>	7 TAG20090501T083307	May 1, 2009 8:33:09 AM	SPFILE, CONTROLFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	6 BACKUP_ORCL_EXAMPL_350109083242	May 1, 2009 8:33:07 AM	DATAFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	5 TAG20090501T082318	May 1, 2009 8:23:19 AM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	4 BACKUP_ORCL_EXAMPL_350109081759	May 1, 2009 8:23:15 AM	ARCHIVED LOG	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	3 TAG20090501T082249	May 1, 2009 8:22:52 AM	CONTROLFILE, SPFILE	DISK	AVAILABLE	NO	1
<input type="checkbox"/>	2 BACKUP_ORCL_EXAMPL_350109081759	May 1, 2009 8:22:43 AM	DATAFILE	DISK	AVAILABLE	NO	1

A backup recorded in the RMAN repository has one of the following status values:

- 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 offsite or on a disk that is currently not mounted)

Backups can also be obsolete. An obsolete backup is, based on the currently configured retention policy, no longer needed to satisfy data recovery goals.

Maintenance tasks that you can perform in Database Control include the following:

- Viewing details about your backups
- Cross-checking your repository, which means checking whether backups listed in the repository exist and are accessible, and marking as expired any backups not accessible at the time of the cross-check
- Deleting the record of expired backups from your RMAN repository
- Deleting obsolete backups from the repository and from the backup media
- Validating backups to ensure that a given backup is available and not corrupted

Note: If a backup no longer exists, then immediately delete the backup record 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 must perform a recovery.

As with the backup and restore commands in Database Control, the commands to cross-check, delete, and change the status of backups are translated to RMAN commands. The commands are submitted as RMAN jobs that can be run immediately or be scheduled. Some tasks, such as periodic cross-checks of your backups, should be among the regularly scheduled components of your backup strategy.

If you use a fast recovery area for backup storage, then many maintenance activities are reduced or eliminated. The automatic disk space management mechanisms delete backups and other files as needed, thereby satisfying disk space demands for ongoing database operations without compromising the retention policy.

Cross-Checking Backups

Cross-checking a backup synchronizes the physical reality of backups with their logical records in the RMAN repository. For example, if a backup on disk was deleted with an operating system command, then a cross-check detects this condition. After the cross-check, the RMAN repository correctly reflects the state of the backups.

Backups to disk are listed as available if they are still 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 on tape. The file headers are not checked for corruption. Backups that are missing or corrupt are listed as expired.

To cross-check individual files:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section, click **Manage Current Backups**.
The Manage Current Backups page appears.
3. Search for the backup sets or image copies whose contents you want to cross-check.
4. In the Results section, select each backup to be included in the cross-check operation.

Database Control does not support selecting both image copies and backup sets for a cross-check within a single operation.

5. Click **Crosscheck** at the top of the Results list.

A confirmation page is displayed. After you click **Yes**, Database Control performs the cross-check.

To cross-check all files:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section, click **Manage Current Backups**.
The Manage Current Backups page appears.

3. In the Manage Current Backups page, click **Crosscheck All**.

The Crosscheck All: Specify Job Parameters page appears. You can schedule the cross-check to run now or later, or specify regularly scheduled cross-checks.

4. Click **Submit Job**.

Note: Cross-checking all backups in the RMAN repository may take a long time, especially for tape backups. A cross-check of all files, unlike cross-checking individual files, is handled as a scheduled job.

See Also:

- ["Managing Backups"](#)
- *Oracle Database Backup and Recovery User's Guide* for details on how to perform this procedure using the RMAN CROSSCHECK command

Deleting Expired Backups

Deleting expired backups removes from the RMAN repository those backups that are listed as EXPIRED. Expired backups are those found to be inaccessible during a cross-check. No attempt is made to delete the files containing the backup from disk or tape; this action updates only the RMAN repository.

To delete expired backups:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section, click **Manage Current Backups**.
The Manage Current Backups page appears.
3. In the Manage Current Backups page, click **Delete All Expired**.
This action deletes expired backup sets and expired image copies from the RMAN repository, regardless of which subpage is currently active.
The Delete All Expired: Specify Job Parameters page appears.
4. Optionally, select **Perform the operation 'Crosscheck All' before 'Delete All Expired'**.
By performing the cross-check immediately before deleting expired backups, RMAN has up-to-date information about which backups are expired.
5. Click **Submit Job**.
A message should appear indicating that your job was successfully submitted.

Marking Backups as Available or Unavailable

If 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 offsite, then you can mark those backups as unavailable. RMAN does not use unavailable backups to restore or recover data.

Note: Backups stored in the fast recovery area cannot be marked as unavailable.

RMAN keeps the record of unavailable backups in the RMAN repository and does not delete backups listed as unavailable when you delete expired backups. If the unavailable backups become accessible again, then you can mark them as available.

To mark backups as available or unavailable:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section, click **Manage Current Backups**.
The Manage Current Backups page appears.
3. In the Search section, find the backups whose status you want to change.
4. Click the **Select** check box next to each backup in the Results list of backups for which you want to change its availability.
5. Do one of the following:
 - Select **Change to Available**.
 - Select **Change to Unavailable**.

Note: If you restricted the backups listed by searching only for available backups, then only the **Change to Unavailable** button appears. If you restricted the backups listed by searching only for unavailable backups, then only the **Change to Available** button appears.

A confirmation message appears.

6. Click **Yes** to perform the change operation.

Deleting Obsolete Backups

This section explains how to delete obsolete backups, which are those no longer needed by the configured retention policy. If you use a fast recovery area as your only disk-based backup destination, then you never have to delete obsolete backups from disk. The fast recovery area keeps files as specified by the retention policy, and deletes them only when space is needed.

To delete obsolete backups:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. In the Manage section, click **Manage Current Backups**.
The Manage Current Backups page appears.
3. Click **Delete All Obsolete**.

All obsolete backups (both backup sets and image copies) are deleted, regardless of whether you clicked **Delete All Obsolete** while viewing the Backup Sets or Image Copies subpage.

The Delete All Obsolete: Specify Job Parameters page appears.

4. In the Schedule section, do one of the following:
 - Select **One Time (Immediately)** to run the deletion job immediately.
 - Schedule the deletion as you would a backup job.
5. Click **Submit Job**.

Performing Oracle Advised Recovery

The Oracle advised recovery feature uses Data Recovery Advisor, which is an Oracle Database feature that automatically diagnoses data failures, determines and presents appropriate repair options, and performs repairs if requested by the user. By providing a centralized tool for automated data repair, Data Recovery Advisor improves the manageability and reliability of an Oracle database.

Note: To recover an Oracle RAC database using Data Recovery Advisor you must use the RMAN interface. You cannot use Enterprise Manager.

About Data Recovery Advisor

In the context of Data Recovery Advisor, a health check is a diagnostic procedure run by the Health Monitor to assess the state of the database or its components. Health checks are invoked reactively when an error occurs. You can also invoke checks manually.

A **failure** is a persistent data corruption detected by a health check. Failures are usually detected reactively. A database operation involving corrupted data results in an error, which automatically invokes a health check in the database. The check searches the database for failures related to the error. If failures are diagnosed, then they are recorded in the Automatic Diagnostic Repository (ADR).

You can use Data Recovery Advisor to generate repair advice and repair failures only after failures have been detected by the database and stored in ADR. Data Recovery Advisor can report on and repair failures such as inaccessible files, physical and logical block corruptions, and I/O failures. Every failure has a failure priority: CRITICAL, HIGH, or LOW. Every failure also has a failure status of OPEN or CLOSED.

You can also use Data Recovery Advisor to view repair options. A **repair** is an action that fixes one or more failures. Examples of repairs include block media recovery, data file media recovery, and Oracle Flashback Database. Typically, Data Recovery Advisor presents both automated and manual repair options. If appropriate, you can choose an automated repair option in order to perform a repair. In this case, Data Recovery Advisor verifies the repair success, and closes the relevant repaired failures.

Using Data Recovery Advisor

The recovery process begins when you either suspect or discover a failure. You can discover failures in many ways, including error messages, alerts, trace files, and health checks. You can then use Data Recovery Advisor to gain information and advice about failures and repair them automatically.

This section describes a scenario in which you use Data Recovery Advisor to repair a corrupted block. Assume that the Health Meter on the Database Home page indicates that an incident has occurred. The Alerts section of the page indicates that a block corruption has occurred.

To use the Oracle advised recovery strategy:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. Click **Perform Recovery** in the Manage section.

The Perform Recovery page appears. The Perform Recovery page is divided into two sections: Oracle Advised Recovery and User-Directed Recovery. The Oracle Advised Recovery section uses Data Recovery Advisor to automate recovery.

The screenshot shows the 'Perform Recovery' page in Oracle Performance Center. At the top, there's an 'Information' section with 'Database Failures - 1'. Below that is the 'Perform Recovery' header. The main content is split into two primary sections: 'Oracle Advised Recovery' and 'User Directed Recovery'. The 'Oracle Advised Recovery' section indicates that the Data Recovery Advisor has detected failures, with a summary of 'Critical: 0 High: 1 Low: 0' and a description of 'One or more non-system datafiles are offline'. There is an 'Advise and Recover' button. The 'User Directed Recovery' section allows for selecting a 'Recovery Scope' (currently 'Whole Database') and an 'Operation Type' (currently 'Recover to the current time or a previous point-in-time'). Other options include 'Restore all datafiles' and 'Recover from previously restored datafiles'. There is a 'Recover' button. Below these are sections for 'Decrypt Backups' and 'Host Credentials', which includes fields for 'Username' (pre-filled with 'oracle') and 'Password', and a checkbox for 'Save as Preferred Credential'. On the right side, there is an 'Overview' sidebar with a list of recovery actions: 'Recover database failures as advised by Oracle', '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 as corrupted, or based on datafile block IDs or tablespace block addresses', and 'Flashback database or tables to a specific system change number (SCN) or timestamp'.

3. Look for failures in the Oracle Advised Recovery section.

In the screenshot shown in Step 2, the section indicates that one failure with high priority exists. The failure indicates that 1 or more data files are offline.

4. Do one of the following:
 - Click **Advise and Recover**.
 - Click the number next to the failure status.

The View and Manage Failures page appears.

5. In the Priority drop-down list, select **All** and then click **Go**.

This action enables you to view all failures known to Data Recovery Advisor. The data failures are represented as a navigation tree at the bottom of the page.

6. Perform the following actions:
 - a. Select **Data Failures** from the navigation tree to expand it (if it is not expanded).

- b. Select the failure to expand it.
- c. Click **Advise**.

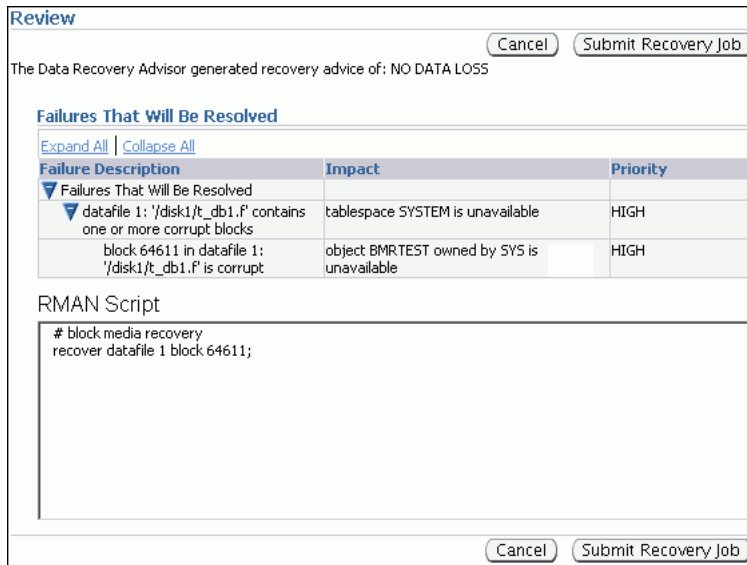
The Recovery Advice page appears.

This page shows the RMAN script that is used to repair the failure. For example, for a corrupted block, the script might look similar to the following:

```
# block media recovery
recover data file 1 block 63467;
```

- 7. Click **Continue**.

The Review page appears. This page summarizes the recovery job.



- 8. Click **Submit Recovery Job**.

The Job Activity page appears.

A table describes details of the repair such as the job name, the job status, the time that the repair is scheduled to be performed, the owner of the job, and so on.

- 9. Click the name of the repair job and click **View Results**.

The Job Run page appears.

You can click the job step in the navigation tree to view the results of the job.

Performing User-Directed Recovery

Oracle Enterprise Manager Database Control (Database Control) User-Directed Recovery provides a Recovery wizard that enables you to use flashback features and perform restore operations and recovery procedures. For example, you can do the following:

- Repair unwanted changes to database objects with the logical flashback features
- Rewind the entire database with Oracle Flashback Database
- Completely restore and recover the database
- Perform point-in-time recovery of the database or selected tablespaces

- Perform block media recovery of data files that have corrupted blocks

Database Control can determine which parts of the database must be restored and recovered, including detecting situations such as corrupted database files before they affect database operations. Database Control guides you through the recovery, prompting you for any required information and performing the specified recovery actions.

This section contains a few typical recovery examples so that you can become familiar with the Perform Recovery page. You can use the Perform Recovery page to access other whole database or object-level recovery features of Database Control.

Rewinding a Table Using Oracle Flashback Table

Oracle Flashback Table enables you to rewind one or more tables back to their contents at a previous time without affecting other database objects. Thus, you can recover from logical data corruptions such as table rows added or deleted accidentally. Unlike point-in-time recovery, the database remains available during the flashback operation.

For this example, you use Flashback Table on the `employees` table in the `hr` schema. Assume that an erroneous update shortly after October 23, 2005 at 15:30:00 has changed the `lastname` column for all employees to an empty string, and you must return the original `lastname` values to the table.

Enabling Row Movement on a Table

Before you can use Flashback Table, you must ensure that row movement is enabled on the table to be *flashed back*, or returned to a previous state. Row movement indicates that rowids will change after the flashback occurs. This restriction exists because if rowids before the flashback were stored by an application, then there is no guarantee that the rowids will correspond to the same rows after the flashback.

To enable row movement on a table:

1. On the Database Home page, click **Schema** to display the Schema subpage.
2. Click **Tables** in the Database Objects section.
The Tables page appears.
3. To find the target table for Flashback Table, do the following:
 - a. Enter the schema name in the **Schema** field and, optionally, the table name in the **Object Name** field.
 - b. Click **Go** to search for the table.
When you search for tables, for example, in the `hr` schema, you may have to page through the search results to find your table.
4. Select the table from the list of tables and click **Edit**.
In this scenario, select `employees`.
The Edit Table: `table_name` page appears.
5. Click **Options** to go to the Options subpage.
6. Complete the following steps:
 - a. Set **Enable Row Movement** to **Yes**.
 - b. Click **Apply** to update the options for the table.

An update message appears.

7. Complete the following steps:
 - a. Click **Tables** at the top of the page to return to the search results.
 - b. Enable row movement on more tables by repeating Step 1 through Step 6 for each table.

For this example, you should also enable row movement on the tables `hr.jobs` and `hr.departments`.

Performing a Flashback Table Operation

In this example, you rewind the `hr.employees` table and its dependent tables to a previous point in time.

To perform the Flashback Table operation:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. Select **Perform Recovery** from the Manage section.

The Perform Recovery page appears.
3. In the User Directed Recovery section, select **Tables** from the Recovery Scope list.

The page reloads with options appropriate for object-level recovery of tables.
4. For **Operation Type**, choose **Flashback Existing Tables**, and click **Recover**.

The Perform Object Level Recovery: Point-in-time page appears.
5. Choose the target time for the Flashback Table operation, and click **Next**.

Note: If you do not know the time at which the unwanted changes occurred, then you can investigate the history of transactions affecting this table by selecting **Evaluate row changes and transactions to decide upon a point in time**. Oracle Flashback Version Query enables you to review all recent changes to the target table. Use of this feature is beyond the scope of this documentation.

For this example, assume that rows were accidentally inserted 5 minutes ago. Select **Flashback to a timestamp** and enter a time 5 minutes before now.

The Perform Object Level Recovery: Flashback Tables page appears.

6. Enter table names in the **Tables to Flashback** text box and then click **Next**.

You can enter multiple table names to flash back several tables to the same time. You can also click **Add Tables** and search for more tables. For this example, enter the text `hr.employees` in the **Tables to Flashback** text box.

If your table has other dependent tables, then the Dependency Options page appears. This page asks how dependencies should be handled.
7. Do one of the following, and then click **Next**:
 - Select **Cascade** to flash back any dependent tables.
 - Select **Restrict** to flash back only the target table.
 - Select **Customize** to choose which dependent tables to flash back and which to leave as they are.

You can click **Show Dependencies** to see which tables will be affected.

Note: Row movement must be enabled on all affected tables, not just the initial target tables.

In this example, the `hr.employees` table has dependent tables `hr.jobs` and `hr.departments`, so select **Cascade** and then click **Next**.

The Perform Object Level Recovery: Review page appears.

8. Click Submit.

When the operation is completed, a Confirmation page displays the results. Click **OK**.

Recovering a Dropped Table Using Oracle Flashback Drop

Oracle Flashback Drop enables you to reverse the effects of dropping (deleting) a table, returning the dropped table to the database along with dependent objects such as indexes and triggers. This feature stores dropped objects in a recycle bin, from which they can be retrieved until the recycle bin is purged, either explicitly or because space is needed.

As with Flashback Table, you can use Flashback Drop while the database is open. Also, you can perform the flashback without undoing changes in objects not affected by the Flashback Drop operation. Flashback Table is more convenient than forms of media 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, you cannot recover tables in the `SYSTEM` tablespaces with Flashback Drop, regardless of the tablespace type.

Dropping a Table

For the purpose of learning about Flashback Drop, you will create a new table named `reg_hist` and then drop it. The database places the table in the recycle bin so that it can be retrieved with the Flashback Drop feature.

To create and then drop a table:

1. On the Database Home page, click **Schema** to display the Schema subpage.
2. Click **Tables**.

The Tables page appears.

3. Enter **hr** in the **Schema** field and **regions** in the **Object Name** field. Click **Go**.

The schema and table are listed in the table at the bottom of the page.

4. In the Actions list, select **Create Like** and click **Go**.

The General subpage of the Create Table page appears.

5. Complete the following steps:

- a. In the **Name** field, enter **reg_hist**.
- b. Deselect **Not Null** for the **REGION_ID** column.

- c. Click **Constraints** to open the Constraints subpage.
The Constraints subpage appears.
 - d. Select each constraint and click **Delete**.
 - e. Click **OK** to create the table.
A confirmation message appears.
6. In the **Object Name** field, enter **reg_hist** and click **Go**.
The Tables page displays information about this table.
 7. Click **Delete with Options** to delete the table.
The Delete With Options page appears.
 8. Select **Delete the table definition, all its data, and dependent objects (DROP)**, and then click **Yes**.
A message confirms that the table was deleted.

Retrieving a Dropped Table

This section assumes that you created and then dropped the `reg_hist` table, as described in "Dropping a Table" on page 9-35. The following procedure retrieves `reg_hist` from the recycle bin.

To perform the Flashback Drop operation:

1. On the Database Home page, click **Availability** to display the Availability subpage.
2. Click **Perform Recovery** in the Backup/Recovery section.
The Perform Recovery page appears.
3. In the User-Directed Recovery section, select **Tables** from the Recovery Scope list.
The page reloads with options appropriate for object-level recovery of tables.
4. Select **Flashback Dropped Tables** for the Operation Type, and then click **Recover**.
The Perform Object Level Recovery: Dropped Objects Selection page appears.
5. Search among the dropped objects in the recycle bin for the objects to retrieve.
For this example, enter **hr** in the **Schema Name** field and **reg_hist** in the **Table** field, and click **Go**.
The Results section lists the objects matching your search. If needed, click the arrow next to Recycle Bin to expand its contents by one level, showing dropped tables matching your search but not their dependent objects.
6. Select each table to retrieve using Flashback Drop.
For this example, select **reg_hist** and then click **Next**.

Note: When a table is retrieved from the recycle bin, all the dependent objects for the table that are in the recycle bin are retrieved with it. They cannot be retrieved separately.

The Perform Object Level Recovery: Rename page appears.

7. If needed, enter new names for dropped objects that you are retrieving. For this scenario, do not specify a new name. Click **Next**.

The primary reason for renaming objects when you retrieve them from the recycle bin is because you might have created new tables with the same names as tables being retrieved. If you must rename some objects, then enter new names as needed in the **New Name** field.

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, and the names they will have when the Flashback Drop operation is complete.

8. Review the changes and then click **Submit**.

A confirmation page should indicate the success of the operation.

9. Click **OK** to return to the Availability page.

Rewinding a Database Using Oracle Flashback Database

Unlike the other flashback features, Oracle Flashback Database operates at a physical level. When you use Flashback Database, your current data files revert to their contents at a previous time. The result is similar to database point-in-time recovery, but Flashback Database can be much faster because it does not require you to restore and recover data files. Also, Flashback Database requires limited application of redo data as compared to media recovery.

Flashback Database uses flashback logs to access previous versions of data blocks and also uses some data in the archived redo log files. To have the option of using Flashback Database to repair your database, you must have configured the database to generate flashback logs as explained in "[Configuring Recovery Settings](#)" on page 9-7.

To perform a Flashback Database operation:

1. On the Database Home page, click **Availability** to display the Availability subpage.

2. Click **Perform Recovery**.

The Perform Recovery page appears.

3. Complete the following steps:

- a. In the User-Directed Recovery section, select **Whole Database**.
- b. Select **Recover to the current time or a previous point-in-time**.
- c. If necessary, provide the host computer credentials.
- d. Click **Recover**.

A Confirmation page appears.

4. Click **Yes** to confirm the shutdown of the database.

The Recovery Wizard page appears. At this point, the shutdown begins.

When the database is shut down and brought to the mounted state, Database Control is also shut down briefly and restarted. During this process, there is a period during which Database Control cannot respond to your browser. Refresh the page until Database Control responds again.

When Database Control has restarted and the database is being started and mounted, Database Control may also briefly report that the database is in the

NOMOUNT state. You are given the choices **Refresh**, **Startup**, and **Perform Recovery**. Refresh the page periodically until the Database Instance page reports that the database instance is mounted before you proceed.

5. Click **Perform Recovery** to resume your recovery session.
6. If you are prompted for host computer and database credentials, then connect with the SYSDBA privilege, or provide host computer credentials for a user in the DBA group.

See "[Specifying Credentials for Backup and Recovery Using Database Control](#)" on page 9-5 for information about credentials.

When the Perform Recovery page is displayed again, it shows that the database is mounted, which is required to restore and recover the entire database.

7. Complete the following steps:
 - a. In the User-Directed Recovery section, select **Whole Database**.
 - b. Select **Recover to the current time or a previous point-in-time**.
 - c. If necessary, provide the host credentials.
 - d. Click **Recover**.

A Confirmation page appears.

8. Complete the following steps:
 - a. Select **Recover to a prior point-in-time**.
 - b. In the **Date** field, select a time 5 minutes before now.
 - c. Click **Next**.

The Perform Whole Database Recovery: Flashback page appears.

9. Select **Yes** to use Flashback Database and then click **Next**.

The Perform Whole Database Recovery: Review page appears.

10. Review the options that you selected and click **Submit**.

When the flashback operation completes, the Perform Recovery: Result page appears.

11. Click **Open Database**.

After the database has opened successfully, click **OK**.

Restoring and Recovering the Database

This section demonstrates how to restore and recover the entire database. This example assumes that you are restoring and recovering your database after the loss of one or more data files, but you still have a usable server parameter file and control file. You can also use Database Control to restore a lost server parameter file or control file.

To restore and recover the entire database:

1. Perform Step 1 through Step 7 from the procedure in "[Rewinding a Database Using Oracle Flashback Database](#)" on page 9-37.
2. Do one of the following:
 - Select **Recover to the current time** to recover all transactions to your database up until now.

- Select **Recover to a prior point-in-time** to recover transactions only up through some point in time in the past.

For this example, select **Recover to the current time**, and then click **Next**.

The Perform Whole Database Recovery: Rename page appears.

3. Select **No** to restore the files to their default locations, and then click **Next**.

The Perform Whole Database Recovery: Review page appears.

4. Click **Submit** to start the restoration and recovery of the database.

Note: In some repair scenarios, such as a complete restoration and recovery of your database, the database state is altered by steps you take while using the wizard. Database Control displays warnings each time a significant database change results from selecting **Continue** during the recovery process. Pay close attention to these warnings.

When the recovery completes, the Perform Recovery: Result page appears.

5. Click **Open Database** and then **OK**.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for more details about point-in-time recovery

Backup and Recovery: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this section, and includes annotated screenshots. To view the Backup and Recovery OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/backup/backup.htm

Monitoring and Tuning the Database

Monitoring the performance of a database and ensuring that it performs optimally is an important task for a database administrator. This chapter discusses the features and functions included in Oracle Database that make it easy to monitor database health, identify performance problems, and implement any corrective actions.

This chapter contains the following topics:

- [Proactive Database Monitoring](#)
- [Diagnosing Performance Problems Using ADDM](#)
- [Using Advisors to Optimize Database Performance](#)
- [Monitoring and Tuning: Oracle By Example Series](#)

Proactive Database Monitoring

Oracle Database makes it easy to monitor the health and performance of your database. It monitors the vital signs (or metrics) related to database health and performance, analyzes the workload running against the database, and automatically identifies any issues that need your attention as an administrator. The identified issues are presented as alerts and performance findings on the Database Home page. You can also configure Oracle Enterprise Manager Database Control (Database Control) to notify you of issues by e-mail.

This section discusses the following topics:

- [About Alerts](#)
- [Performance Self-Diagnostics: Automatic Database Diagnostic Monitor](#)
- [Monitoring General Database State and Workload](#)
- [Managing Alerts](#)

About Alerts

Alerts help you monitor your database. Most alerts notify you of when particular metric thresholds are exceeded. For each alert, you can set critical and warning threshold values. These threshold values are meant to be boundary values that when exceeded, 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 Database generates 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 Database issues several alerts, including the following:

- Archive Area Used (warning at 80 percent full)
- Broken Job Count and Failed Job Count (warning when goes above 0)
- Current Open Cursors Count (warning when goes above 1200)
- Dump Area Used (warning at 95 percent full)
- Session Limit Usage (warning at 90 percent, critical at 97 percent)
- Tablespace Space Used (warning at 85 percent full, critical at 97 percent full)

You can modify these alerts and others by setting their metrics.

For more information, see ["Managing Alerts"](#) on page 10-8.

Performance Self-Diagnostics: Automatic Database Diagnostic Monitor

Oracle Database includes a self-diagnostic engine called Automatic Database Diagnostic Monitor (ADDM). ADDM makes it possible for 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. **Snapshots** are sets of historical data for specific time periods that are used for performance comparisons by ADDM. The default collection interval for a snapshot is one hour. Snapshots provide a statistical summary of the state of the system at a point in time. These snapshots are stored in Automatic Workload Repository (AWR), residing in the `SYSAUX` tablespace. The snapshots are stored in this repository for a set time (8 days by default) before they are purged to make room for new snapshots.

ADDM analyzes data captured in AWR to determine the major problems in the system, and usually recommends solutions and quantifies expected benefits. ADDM analysis results are represented as a set of **findings**.

Generally, the performance problems that ADDM can call attention to include the following:

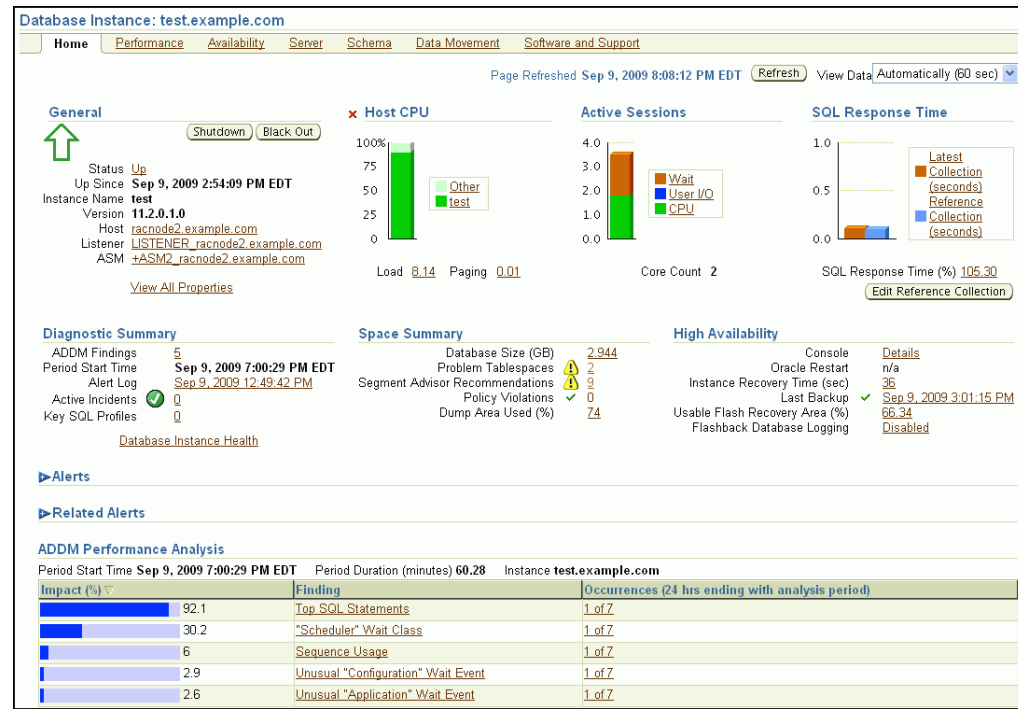
- Resource contention (bottlenecks), such as when your database is using large amounts of CPU time or memory due to high load SQL statements
- Poor connection management, such as when your application is making too many logins to the database
- Lock contention in a multiuser environment, such as when one user process acquires a lock to safely update data in a table, causing other user processes that must acquire locks against the same table to wait, resulting in a slower database performance

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide* for more information about automatic database performance monitoring
- ["Diagnosing Performance Problems Using ADDM"](#) on page 10-15

Monitoring General Database State and Workload

The Database Home page enables you to monitor the state and workload of your database. It provides a central place for general database state information and is updated periodically.



To monitor the general database state and workload:

- Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
- (Optional) Click the **Refresh** button to update the information displayed.
By default, the Database Home page automatically refreshes every 60 seconds. You can prevent automatic refresh by selecting **Manually** in the View Data list at the top right-hand corner of the page. You must then click **Refresh** to view the latest information.
The date and time that data was last collected from the database is displayed to the left of the Refresh button.
- Get a quick overview of the database state in the General section, which includes the following information:
 - Status of the database instance, Up or Down
Click the **Status** link to drill down to database availability details.
 - Time the database was last started
 - Instance name
 - Oracle Database version
 - Host name
Click the **Host** link to drill down to host details.

- Listener name

Click the **Listener** link to drill down to listener details.

Click **View All Properties** to see the Oracle home path and whether the database is read-only or read/write.
- 4. View CPU utilization in the Host CPU section, which includes the following information:
 - Bar chart

This chart shows the percentage of CPU time used by the database and other processes. The chart legend contains links for the database instance and for other CPU processes.

Click the **Other** link in the chart legend to see how the utilization of CPU, memory, and disk I/O change over time.

Click the instance name link in the chart legend to see the Top Activity page. It includes a graph of active sessions over time, details about SQL statements issued, and the most active sessions.
 - CPU load

This is the average number of processes waiting to be scheduled for the CPU in the previous minute.

Click the **Load** link to see how the utilization of CPU, memory, and disk I/O change over time.
 - Paging

This is the number of memory pages (fixed-length block of instructions, data, or both) that are paged out (moved out of active memory) each second.

Click the **Paging** link to see how the utilization of CPU, memory, and disk I/O change over time.
- 5. Investigate the Active Sessions section, where you can further explore the cause of performance problems, such as your database taking up most of the CPU time on the server. This section displays a bar graph with the following information:
 - Waits

This is the value for all wait classes combined, excluding user I/O and idle wait events. **Wait classes** are groupings of wait events based on the type of wait.

If other processes are taking up most of your CPU time, then this indicates that some other application running on the database host computer could be causing performance problems.

Click the **Wait** link to go to the Performance page to view potential problems inside and outside the database.
 - User I/O

This is the average number of active sessions waiting for user I/O. User I/O means that the workload originating from the user causes the database to read data from disk or write data to disk.

Click the **User I/O** link to go to the Performance page to view potential problems inside and outside the database.
 - CPU

This is the average active sessions using CPU.

Click the **CPU** link to see a chart showing more detailed information about active sessions over time.

6. View the Diagnostic Summary section, which includes the following information:

- ADDM Findings

This shows the count of ADDM findings from the most recent ADDM run. Click the number adjacent to the ADDM Findings link to go to the ADDM page.

- Period Start Time

This is the start time of the time period most recently analyzed by ADDM. It is shown only if there are ADDM findings.

- Alert Log

This is the timestamp of the most recent alert log entry that describes an ORA-error.

Click the **Alert Log** link to go to the Alert Log Errors page, which shows a list of log entries that contain errors.

- Active Incidents

This shows the count of active *incidents*, which are occurrences of critical errors in the database. You are encouraged to investigate critical errors and report them to Oracle Support Services. Click the count to go to the Support Workbench home page.

- Database Instance Health

Click **Database Instance Health** to display the Database Instance Health page, which includes graphical timelines of incidents, ADDM findings, and alerts. You can use these graphs for identifying correlations between incidents and alerts generated and performance issues on the system.

7. View the SQL Response Time section.

This is the current response time of a tracked set of SQL statements as compared to the response time for the reference collection. A **reference collection**, or SQL Tuning Set, is set of SQL statements that represents the typical SQL workload on your production system. If the current response time and reference collection response time are equal, then the system is running as it should. If the current response time is greater than the reference collection response time, then one or more SQL statements are performing more slowly than they should. The lower the current response time, the more efficiently the tracked SQL statements run.

Click the **SQL Response Time** link to see response time metrics for the previous 24 hours. If the reference collection is empty, then click **Reset Reference Collection** to go to a page where you can create a reference collection.

8. View the Space Summary section.

If the number adjacent to the Segment Advisor Recommendations label is not zero, then it means the Segment Advisor has identified candidate segments for space defragmentation. Click the number to view recommendations for how to defragment these segments.

9. View the Alerts section, which includes the following items:

- Category list

Optionally choose a category from the list to view alerts only in that category.

- Critical

This is the number of metrics that have exceeded critical thresholds plus the number of other critical alerts, such as those caused by incidents (critical errors).

- Warning

This is the number of metrics that have exceeded warning thresholds.

- Alerts table

Click the message to learn more about the alert.

10. View the ADDM Performance Analysis section, if present. This section contains the following items:

- Period Start Time

This is the start time of the period most recently analyzed by ADDM.

- Period Duration in minutes

This is the duration of the period most recently analyzed by ADDM.

- Instance name

- ADDM findings table

This table lists the ADDM findings, their estimated impact on database performance, a description of the finding, and the number of times the finding occurred in snapshots collected during the previous 24 hours. For example, a finding with Occurrences listed as 34 of 43 has occurred in 34 of the last 43 snapshots.

Click the finding to view finding details, to view recommendations, and in some cases to implement recommendations or start advisors.

ADDM Performance Analysis		
Period Start Time Jun 24, 2009 9:35:50 AM PDT		
Period Duration (minutes) 1.52		Instance em112
Impact (%) ▾	Finding	Occurrences (24 hrs ending with analysis period)
47.6	Top SQL Statements	11 of 630
42	"Scheduler" Wait Class	12 of 630
7.6	Session Connect and Disconnect	5 of 630
3.6	Buffer Busy	4 of 630

To view database performance over time:

1. At the top of the Database Home page, click **Performance**.

The Performance page appears, displaying a summary of CPU utilization, average active sessions, instance disk I/O, and instance throughput for the recent time period.

2. Use the Additional monitoring links to drill down to Top Activity and other data.

The types of actions you can take to improve host performance depends on your system, and can include eliminating unnecessary processes, adding memory, or adding CPUs.

Managing Alerts

The following topics describe how to manage alerts:

- [Viewing Metrics and Thresholds](#)
- [Setting Metric Thresholds](#)
- [About Responding to Alerts](#)
- [Clearing Alerts](#)
- [Setting Up Direct Alert Notification](#)

Viewing Metrics and Thresholds

To effectively diagnose performance problems, statistics must be available. Oracle generates many types of cumulative statistics for the system, sessions, and individual SQL statements. Oracle also tracks cumulative statistics on segments and services. A **metric** is defined as the rate of change in some cumulative statistic. Metrics are computed and stored in Automatic Workload Repository, and are displayed on the All Metrics page, which can be viewed by clicking **All Metrics** under Related Links on the Database Home page.


To view metrics for your database:

1. On the Database Home page under Related Links, click **All Metrics**.

The All Metrics page appears.

2. Click a specific metric link.

A details page appears, with more information about the metric. Online Help for this page describes the metric.

All Metrics				
Collected From Target Jun 23, 2009 12:19:28 PM PDT 				
Expand All Collapse All				
Metrics	Thresholds	Collection Schedule	Upload Interval	Last Upload
▼ em112				
▶ Archive Area	Some	Every 15 Minutes	Every Collection	Jun 23, 2009 12:19:16 PM
▶ Data Failure	Some	Every 5 Minutes	Every Collection	-
▶ Database Files	None	Server Generated	Server Generated	Server Generated
▶ Database Job Status	All	Every 5 Minutes	On Alert	Jun 23, 2009 10:55:43 AM
▶ Database Limits	Some	Server Generated	Server Generated	Server Generated
▶ Database Replay	None	Real-time Only	n/a	n/a
▶ Database Replay Client	None	Server Generated	Server Generated	-
▶ Database Services	None	Server Generated	Server Generated	Server Generated
▶ Database Vault Attempted Violations - Command Rules	None	Every 1 Hour	On Alert	-
▶ Database Vault Attempted Violations - Realms	None	Every 1 Hour	On Alert	-
▶ Database Vault Configuration Issues - Command Rules	All	Every 1 Hour	Every Collection	-
▶ Database Vault Configuration Issues - Realms	All	Every 1 Hour	Every Collection	-
▶ Database Vault Policy Changes	All	Every 1 Hour	Every Collection	-
▶ Deferred Transactions	All	Every 5 Minutes	On Alert	Jun 23, 2009 10:55:19 AM
▶ Dump Area	Some	Every 15 Minutes	Every Collection	Jun 23, 2009 12:19:16 PM
▶ Efficiency	None	Server Generated	Server Generated	Server Generated
▶ Failed Logins	All	Every 30 Minutes	Every Collection	Jun 23, 2009 11:32:24 AM

For each of these metrics, you are able to define warning and critical threshold values, and whenever the threshold is exceeded, Oracle Database issues an alert. Alerts are displayed on the Database Home page under the Alerts heading (or Related Alerts for nondatabase alerts such as a component of Oracle Net).

The following sample output shows two warning alerts for the threshold Tablespace Space Used.

Alerts					
Category		All	Go	Critical ✖ 1	Warning ⚠ 4
Severity	Category	Name	Impact	Message	Alert Triggered
✖	Incident	Generic Internal Error		Internal error (ORA-600[15700]) detected in /u01/app/oracle/diag/rdbms/test/test/alert/log.xml at time/line number: Wed Sep 9 12:27:22 2009:8777.	Sep 9, 2009 12:27:48 PM
⚠	Response	User Logon Time (msec)		User logon time is 6879.37 msec.	Sep 9, 2009 8:01:32 PM
⚠	User Audit	Audited User		User SYS logged on from racnode2.example.com.	Sep 3, 2009 3:49:42 PM
⚠	Tablespaces Full	Tablespace Space Used (%)		Tablespace TEST_MSSM_NEW is 86 percent full	Sep 8, 2009 12:18:11 PM
⚠	Tablespaces Full	Tablespace Space Used (%)		Tablespace TEST_ASSM is 87 percent full	Sep 8, 2009 12:18:11 PM

Setting thresholds is discussed in ["Setting Metric Thresholds"](#) on page 10-11. Actions you might take to respond to alerts are discussed in ["About Responding to Alerts"](#) on page 10-13.

When the condition that triggered the alert is resolved and the metric value is no longer outside the boundary, Oracle Database 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 Database advisors.

Setting Metric Thresholds

Oracle Database provides a set of predefined metrics, some of which have predefined thresholds. There may be times when you want to set thresholds for other metrics, or you want to alter existing threshold settings.

["Changing Space Usage Alert Thresholds for a Tablespace"](#) on page 6-19 describes one means of setting a threshold, 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.

To set metric thresholds:

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. Under the Related Links heading, click **Metric and Policy Settings**.

The Metric and Policy Settings page appears.

This page displays the existing thresholds for metrics and any response actions that have been specified.

Metric and Policy Settings						
Metric Thresholds						Policies
View: Metrics with thresholds						
Metric	Comparison Operator	Warning Threshold	Critical Threshold	Corrective Actions	Collection Schedule	Edit
Access Violation	Matches		.*	None	Every 5 Minutes	
Access Violation Status	>		0	None	Every 5 Minutes	
Archive Area Used (%)	>	80		None	Every 15 Minutes	
Archiver Hung	Matches		.*	None	Every 5 Minutes	
Archiver Hung Status	>		0	None	Every 5 Minutes	
Audited User	=	SYS		None	Every 15 Minutes	
Average Users Waiting Count						
Administrative	>	10		None		
Application	>	10		None		
Cluster	>	30		None		
Commit	>	30		None		

3. In the View list, do one of the following:
 - Select **Metrics with thresholds** to view only those metrics with thresholds, either predefined by Oracle or previously set by you.
 - Select **All Metrics** to view all metrics, whether they have thresholds defined.
4. To set or modify a warning threshold for a particular metric, enter the value you want in the **Warning Threshold** field for that metric.
5. To set or modify a critical threshold for a particular metric, enter the value you want in the **Critical Threshold** field for that metric.
6. To disable or reenable metric collection for a particular metric, or to change its collection schedule, complete the following steps:
 - a. Click the Collection Schedule link for the metric.
The Edit Collection Settings page for that metric appears.
 - b. Click **Disable** to disable collection for this metric, or click **Enable** to enable it.
 - c. If you chose Enable, then you must choose the **Frequency Type** for the Collection Schedule and enter a numeric value in the **Repeat Every** field.
If you chose Disable, then skip this step.
 - d. Do one of the following:
 - Click **Continue** to save your choices and return to the Metric and Policy Settings page.
 - Click **Cancel** to return to the Metric and Policy Settings page without saving your choices.
7. Click a single-pencil icon to use the Edit Advanced Settings page to make changes to Corrective Actions, (Monitoring) Template Override, and Advanced Threshold Settings. Click **Help** on this page for more information about these options.
8. Click a triple-pencil icon to set different threshold values for different instances of the object type being measured.

For example, for each tablespace you can set different warning and critical levels for the Tablespace Space Used metric.

9. Do one of the following:
 - Click **OK** to save your changes and return to the Database Home page.
 - Click **Cancel** to return to the Database Home page without saving your changes.

About 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 activity.

For example, if you receive a Tablespace Space Usage alert, then 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 create available (free) space. See ["Reclaiming Unused Space"](#) on page 6-22.

Additionally, as a response, you can set a corrective script to run as described in ["Setting Metric Thresholds"](#) on page 10-11.

Clearing Alerts

Most alerts are cleared (removed) automatically when the cause of the problem disappears. Other alerts, such as Generic Alert Log Error, are sent to you for notification and must be acknowledged by the database 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 can be viewed from the Database Home page under Related Links. Purging an alert removes it from the Alert History.

To clear or purge an alert:

1. On the Database Home page under Diagnostic Summary, click the **Alert Log** link.
The Alert Log Errors page appears.
2. From the View Data list, select the period for which you want information.
3. Click **Refresh** to refresh the page with the latest information.
4. Do one of the following:
 - Click **Show Open Alerts** to hide alerts that have been cleared.
 - Click **Show Open and Cleared Alerts** to see all alerts.

Note: You will see only one or the other of these buttons, depending on what is currently displayed.

5. Select one or more alerts by clicking their **Select** options.
6. Click **Clear** to clear the alert.
7. Click **Purge** to purge the alert.
8. Click **Clear Every Open Alert** to clear all open alerts.
9. Click **Purge Every Alert** to purge all alerts.

Setting Up Direct Alert Notification

Database Control displays all alerts on the Database Home page. However, you can optionally specify that Database Control provide you direct notification when specific alerts arise. For example, if you choose 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 message similar to the following:

```
Host Name=mydb.us.example.com
Metric=Response Time per Call
Timestamp=08-NOV-2006 10:10:01 (GMT -7:00)
Severity=Critical
Message=Response time per call has exceeded the threshold. See the
latest ADDM analysis.
Rule Name=
Rule Owner=SYSMAN
```

The e-mail message 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 Space Used are set up for notification. However, to receive these notifications, you must set up your e-mail information.

To set up your e-mail information:

1. From any Database Control page, click the **Setup** link, which is visible in the header and footer areas.
2. On the Setup page, select **Notification Methods**.
3. Enter the required information into the Mail Server section of the Notifications Methods page. Click **Help** at the bottom of the page for assistance.

There are other methods of notification, including scripts and SNMP (Simplified Network Management Protocol) program interrupts (traps). The latter can be used to communicate with third-party applications.

At this point, 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 areas.
5. On the Preferences page, select **General**. Click **Add Another Row** in the E-mail Addresses section to enter your e-mail address.
6. Click **Test** to verify that e-mail messages can be sent using the specified information. After the test completes, click **OK**.
7. (Optional) To edit notification rules, for example to change the severity state for receiving notification, select **Rules** under the heading Notification on the left-hand side of the page.

The Notification Rules page appears. Click **Help** for more information about this page.

See Also:

- *Oracle Enterprise Manager Advanced Configuration*, for more information about configuring notification rules
- "[Viewing Metrics and Thresholds](#)" on page 10-8
- "[Setting Metric Thresholds](#)" on page 10-11

Diagnosing Performance Problems Using ADDM

At times, database performance problems arise that require your diagnosis and correction. Usually, these problems are brought to your attention by Automatic Database Diagnostic Monitor (ADDM), which does a top-down system analysis every hour by default, and reports its most significant findings on the Database Home page.

This section contains the following topics:

- [Viewing a Summary of ADDM Performance Findings](#)
- [Responding to ADDM Performance Findings](#)
- [Running ADDM Manually](#)
- [Modifying AWR Snapshot Frequency and Retention](#)

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*

Viewing a Summary of ADDM Performance Findings

ADDM analysis results consist of a description of each finding and a recommended action. You can view a summary of findings and their impacts on the system.

To view a summary of ADDM performance findings:


1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. In the Diagnostic Summary section of the Database Home page, locate the numeric link next to the ADDM Findings label.

Diagnostic Summary

ADDM Findings [4](#)

Period Start Time **Jun 24, 2009 9:35:50 AM PDT**

Alert Log [No ORA- errors](#)

Active Incidents  [0](#)

Key SQL Profiles [0](#)

[Database Instance Health](#)

The number indicates the number of findings from the most recent ADDM run. If this number is zero, then there are no ADDM findings to investigate.

If this number is not zero, then continue to Step 3.

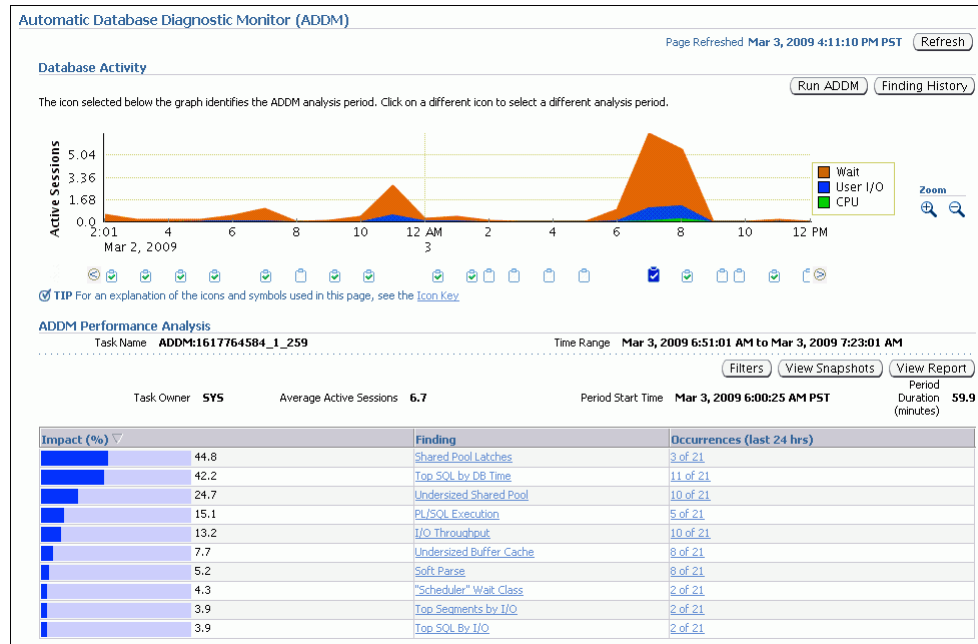
3. View the ADDM Performance Analysis section, below the Alerts section of the Database Home page.

ADDM Performance Analysis		
Period Start Time Jun 24, 2009 9:35:50 AM PDT		Period Duration (minutes) 1.52
Instance em112		
Impact (%) ▾	Finding	Occurrences (24 hrs ending with analysis period)
47.6	Top SQL Statements	11 of 630
42	"Scheduler" Wait Class	12 of 630
7.6	Session Connect and Disconnect	5 of 630
3.6	Buffer Busy	4 of 630

This section is present only if there are ADDM findings. It displays the first five findings from the most recent ADDM run. If there are more than five findings, then controls appear to enable you to view the next five findings, and so on.

- In the Diagnostic Summary section of the Database Home page, locate the nonzero numeric link next to the ADDM Findings label, and click it.

The ADDM page appears, showing a summary of all findings from the most recent run, and a graphical timeline showing session activity over the past 24 hours.



Click **Help** for more information about this page.

Each clipboard icon beneath the graph represents a time range within the 24 hours. Click an icon to view findings for the ADDM run that took place during that icon's time range. If more than one ADDM run occurred during that time range, then you can select an individual ADDM run from the list that appears next to the Task Name label.

See Also:

- Oracle Database 2 Day + Performance Tuning Guide

Responding to ADDM Performance Findings

You can act upon the recommendations that accompany ADDM performance findings.

To respond to ADDM performance findings:

- In the ADDM Performance Analysis section of either the Database Home page or the ADDM page, in the **Finding** column, click a finding.

The Performance Finding Details page appears.

Performance Finding Details: Top SQL Statements

Finding **SQL statements consuming significant database time were found. These statements offer a good opportunity for performance improvement.** [Finding History](#)

Impact (Active Sessions) **4.6**

Percentage of Finding's Impact (%) 47.6

Period Start Time **Jun 24, 2009 9:35:50 AM PDT**

Period Duration (minutes) **1.5**

Filtered **No** [Filters](#)

Recommendations

[Schedule SQL Tuning Advisor](#)

[Select All](#) | [Select None](#) | [Show All Details](#) | [Hide All Details](#)

Select Details	Category	Benefit (%)
<input type="checkbox"/>	SQL Tuning	20.6
Action	Run SQL Tuning Advisor on the SELECT statement with SQL_ID "axabnfyfp4r3p". Additionally, investigate this statement for possible performance improvements. You can supplement the information given here with an ASH report for this SQL_ID. Run Advisor Now Filters SQL Text <code>SELECT E.LAST_NAME, J.JOB_TITLE, D.DEPARTMENT_NAME FROM HR.EMPLOYEES E, HR.DEPAR...</code> SQL ID <code>axabnfyfp4r3p</code>	
Action	Use bigger fetch arrays while fetching results from the SELECT statement with SQL_ID "axabnfyfp4r3p". SQL Text <code>SELECT E.LAST_NAME, J.JOB_TITLE, D.DEPARTMENT_NAME FROM HR.EMPLOYEES E, HR.DEPAR...</code> SQL ID <code>axabnfyfp4r3p</code>	
Rationale	The SQL spent 30% of its database time on CPU, I/O and Cluster waits. This part of database time may be improved by the SQL Tuning Advisor. Look at data given below and an ASH report for further performance improvements.	
Rationale	Database time for this SQL was divided as follows: 100% for SQL execution, 0% for parsing, 0% for PL/SQL execution and 0% for Java execution.	
Rationale	SQL statement with SQL_ID "axabnfyfp4r3p" was executed 35402 times and had an average elapsed time of 0.0062 seconds.	
Rationale	Waiting for event "resmgr:cpu quantum" in wait class "Scheduler" accounted for 53% of the database time spent in processing the SQL statement with SQL_ID "axabnfyfp4r3p".	
Rationale	Top level calls to execute the PL/SQL statement with SQL_ID "3575p6hgzg9gat" are responsible for 100% of the database time spent on the SELECT statement with SQL_ID "axabnfyfp4r3p". SQL Text <code>DECLARE n number; v_employees_last_name ...</code> SQL ID <code>3575p6hgzg9gat</code>	
<input type="checkbox"/>	Show SQL Tuning	11.1
<input type="checkbox"/>	Show SQL Tuning	7.9
<input type="checkbox"/>	Show SQL Tuning	7.9

- In the Recommendations section, click **Show All Details**.
- Choose a finding to respond to, and follow the recommended action. A recommendation can include running an advisor, which you can do by clicking **Run Advisor Now**.
- (Optional) Select one or more findings, and then click the action button above the table of findings. An example of an action button might be Schedule SQL Tuning Advisor.

See Also:

- Oracle Database 2 Day + Performance Tuning Guide

Running ADDM Manually

By default, Oracle Database runs ADDM every hour, immediately after an AWR snapshot. Performance findings from the most recent ADDM run, if any, are listed on the Database Home page. See "Performance Self-Diagnostics: Automatic Database Diagnostic Monitor" on page 10-2 for more information.

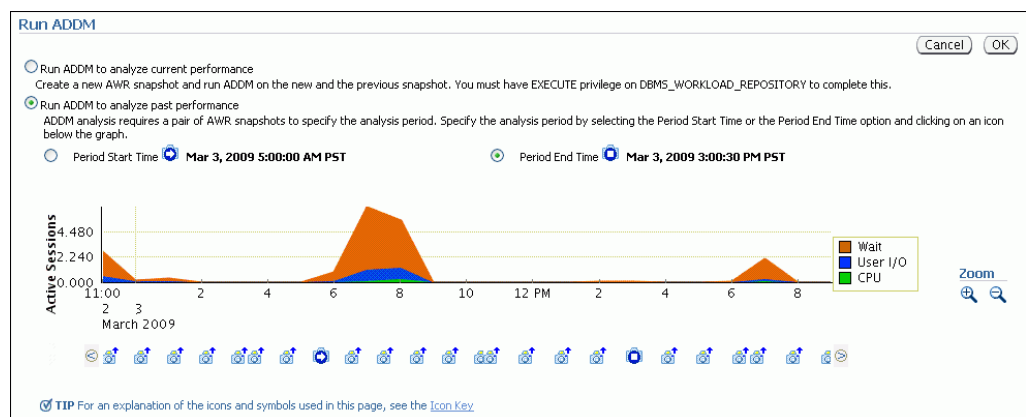
You can also run ADDM manually. Reasons for doing so include running it as a recommended action associated with an alert or running it across multiple snapshots.

Note: If you need more frequent ADDM reporting, then you can also modify the default snapshot interval. To do so, see "[Modifying AWR Snapshot Frequency and Retention](#)" on page 10-22.

The following steps describe how to run ADDM to analyze a period that spans multiple snapshots.

To run ADDM manually:

1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. Under Related Links at the bottom of the page, click **Advisor Central**.
The Advisor Central page appears.
3. Click **ADDM**.
The Run ADDM page appears.



4. Select **Run ADDM to analyze past performance**.
5. Complete the following steps to choose a period start time:
 - a. Select the **Period Start Time** option.
 - b. (Optional) Examine the timeline graph and note the number of snapshots available in each 2-hour period. Each snapshot is represented by a camera icon. If your snapshots occur more frequently than once each hour and you want to view the timeline of snapshots with more precision, then click the zoom-in icon at the right. (The zoom-in icon is a magnifying glass with a plus-sign.)
 - c. (Optional) Below the timeline graph, click the left-arrow icon or right-arrow icon at the left and right sides of the page to change the period shown in the timeline graph.
 - d. Click a camera icon.
A bold right arrow indicating period start time covers the camera icon, and the Period Start Time field above the timeline graph changes to the selected date and time.
6. If you want a period end time other than the default (the most recent snapshot), then complete the following steps:

- a. Above the timeline graph, select the **Period End Time** option.
 - b. (Optional) Click the left-arrow icon or right-arrow icon at the left and right sides of the page to change the period shown in the timeline graph.
 - c. Click a camera icon.
A bold square (like a stop button on a recording device) covers the camera icon, and the Period End Time field changes to the selected date and time.
7. Click **OK** to start the analysis.
- After a short delay, the Automatic Database Diagnostic Monitor page appears, showing its findings.
8. (Optional) In the Task Name list, note the task name.
- A typical task name is "Task_52." You can return to this analysis at a later time by searching for and selecting the task name in the Advisor Central page.

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*

Modifying AWR Snapshot Frequency and Retention

ADDM analysis is based on AWR snapshots, which have a default frequency of once each hour and a default retention period of 8 days.

Note: The snapshot retention period does not apply to preserved snapshots, which are retained forever. A preserved snapshot set is typically created to define a reference period for performance analysis.

You can view and alter these AWR snapshot settings.

To modify AWR snapshot frequency and retention:

1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. At the top of the page, click **Server** to display the Server page.
3. In the Statistics Management section, click **Automatic Workload Repository**.
The Automatic Workload Repository page appears.
4. Click **Edit**.
The Edit Settings page appears.
5. Enter new snapshot retention settings or new snapshot collections settings, and then click **OK**.

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*

Using Advisors to Optimize Database Performance

Oracle Database includes a set of *advisors* to help you manage and tune your database. This section contains background information about these advisors and instructions for their use. The following topics are covered:

- [About Advisors](#)
- [About the SQL Advisors](#)
- [About the Automatic SQL Tuning Advisor](#)
- [Configuring the Automatic SQL Tuning Advisor](#)
- [Viewing Automatic SQL Tuning Results](#)
- [Running the SQL Tuning Advisor](#)
- [Running the SQL Access Advisor](#)
- [Optimizing Memory Usage with the Memory Advisors](#)

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*

About Advisors

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 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. You run advisors from Oracle Enterprise Manager Database Control (Database Control). Some are also run automatically during maintenance windows (time periods).

Advisors are provided to help you improve database performance. These advisors include Automatic Database Diagnostic Monitor (ADDM), SQL advisors, and memory advisors. For example, the Shared Pool Advisor, graphically displays the impact on performance of changing the size of this component of the System Global Area (SGA).

You can run a performance 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 a 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 (Program Global Area).

You can run all advisors from the Advisor Central home page, accessible through a link on the Database Home page. You can also invoke many of the advisors from the Performance page, through recommendations from ADDM, or from alerts.

[Table 10–1](#) describes the performance advisors. Other advisors are listed in [Table 10–2](#) on page 10-24.

Table 10–1 Performance Advisors

Advisor	Description
Automatic Database Diagnostic Monitor (ADDM)	<p>ADDM makes it possible for Oracle Database to diagnose its own performance and determine how any identified problems can be resolved.</p> <p>See "Performance Self-Diagnostics: Automatic Database Diagnostic Monitor" on page 10-2 and "Diagnosing Performance Problems Using ADDM" on page 10-15.</p>
SQL Advisors <ul style="list-style-type: none"> ■ SQL Tuning Advisor ■ SQL Access Advisor 	<p>The SQL Tuning Advisor analyzes one or more SQL statements and makes recommendations for improving performance. This advisor is run automatically during the maintenance periods, but can also be run manually. See "About the Automatic SQL Tuning Advisor" on page 10-26 and "Running the SQL Tuning Advisor" on page 10-33.</p> <p>The SQL Access Advisor tunes a schema to a given SQL workload. For example, the SQL Access Advisor can provide recommendations for creating indexes, materialized views, or partitioned tables for a given workload. See "Running the SQL Access Advisor" on page 10-35.</p>
Memory Advisors <ul style="list-style-type: none"> ■ Memory Advisor ■ SGA Advisor ■ Shared Pool Advisor ■ Buffer Cache Advisor ■ PGA Advisor 	<p>The Memory Advisors provide graphical analyses of total memory target settings, SGA and PGA target settings, or SGA component size settings. You use these analyses to tune database performance and for what-if planning. Depending on the current memory management mode, different memory advisors are available.</p> <ul style="list-style-type: none"> ■ If Automatic Memory Management is enabled, then only the Memory Advisor is available. This advisor provides advice for the total memory target for the instance. ■ If Automatic Shared Memory Management is enabled, then the SGA Advisor and PGA Advisor are available. ■ If Manual Shared Memory Management is enabled, then the Shared Pool Advisor, Buffer Cache Advisor, and PGA Advisor are available. <p>See "Optimizing Memory Usage with the Memory Advisors" on page 10-38. See "Managing Memory" on page 5-15 for more information about memory management modes.</p>

Table 10–2 Other Advisors

Advisor	Description
Segment Advisor	<p>The Segment Advisor provides advice on whether a segment is a good candidate for a shrink operation based on the level of space fragmentation within that segment. 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 Unused Space" on page 6-22.</p>
Undo Advisor	<p>The Undo Advisor 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 Oracle Flashback requirements. See "Computing the Minimum Undo Tablespace Size Using the Undo Advisor" on page 6-34.</p>

See Also:

- *Oracle Database 2 Day + Performance Tuning Guide*

About the SQL Advisors

The SQL advisors examine a given SQL statement or a set of SQL statements and provide recommendations to improve efficiency. These advisors can make various types of recommendations, such as creating **SQL profiles** (a collection of information that enables the query optimizer to create an optimal execution plan for a SQL statement), restructuring SQL statements, creating additional indexes, materialized views, or partitions, and refreshing optimizer statistics. Oracle Enterprise Manager Database Control (Database Control) enables you to accept and implement many of these recommendations with just a few mouse clicks.

The two SQL advisors are the SQL Tuning Advisor and the SQL Access Advisor.

SQL Tuning Advisor

You use the SQL Tuning Advisor to tune a single or multiple SQL statements. Typically, you run this advisor in response to an ADDM performance finding that recommends its use. You can also run it periodically on the most resource-intensive SQL statements, and on a SQL workload.

When tuning multiple SQL statements, the SQL Tuning Advisor does not recognize interdependencies between the SQL statements. It solves SQL performance problems by identifying problems with individual SQL statements, such as a poorly performing optimizer plan or the mistaken use of certain SQL structures.

You can run the SQL Tuning Advisor against 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.
- **Historical SQL**— A set of SQL statements over any 24 hour window (time period). Use this option for proactive 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.

SQL Access Advisor

The SQL Access Advisor is primarily responsible for making schema modification recommendations. It can recommend that you create access structures such as indexes and materialized views to optimize SQL queries. It can also recommend that you partition tables, indexes, or materialized views to improve query performance.

The SQL Access Advisor takes a SQL workload as input. 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 advisor then makes recommendations to improve the performance of the workload as a whole.

Statement Tuning and Workload Tuning

Note that both the SQL Tuning Advisor and the SQL 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 insert, update, and delete operations. The SQL Tuning advisor does not take this into account while generating new index recommendations.

The SQL Access Advisor, however, 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, then the new index is not 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:

- *Oracle Database 2 Day + Performance Tuning Guide* for more information about tuning SQL statements with the SQL Tuning Advisor
- *Oracle Database 2 Day + Performance Tuning Guide* for more information about the SQL Access Advisor
- ["Running the SQL Tuning Advisor"](#) on page 10-33
- ["Running the SQL Access Advisor"](#) on page 10-35

About the Automatic SQL Tuning Advisor

Beginning with Oracle Database 11g, the SQL Tuning Advisor runs automatically during system maintenance windows (time periods) as a maintenance task. During each automatic run, the advisor selects high-load SQL queries in the system and generates recommendations on how to tune these queries.

The Automatic SQL Tuning Advisor can be configured to automatically implement SQL profile recommendations. A **SQL profile** contains additional SQL statistics that are specific to the SQL statement and enable the query optimizer to generate a significantly better execution plan at run time. If you enable automatic implementation, then the advisor creates SQL profiles for only those SQL statements where the performance increase would be at least threefold. Other types of recommendations, such as the creation of new indexes, refreshing optimizer statistics, or restructuring SQL, can be implemented only manually. DML statements are not considered for tuning by the Automatic SQL Tuning Advisor.

You can view a summary of the results of automatic SQL tuning over a specified period (such as the previous 7 days), and a detailed report about recommendations made for all SQL statements that the SQL Tuning Advisor has processed. You can then implement selected recommendations. You can also view the recommendations that were automatically implemented.

You can control when the Automatic SQL Tuning Advisor runs, and you can disable it if desired.

See Also:

- ["Viewing Automatic SQL Tuning Results"](#) on page 10-30
- ["Configuring the Automatic SQL Tuning Advisor"](#) on page 10-27
- *Oracle Database 2 Day + Performance Tuning Guide*
- *Oracle Database Administrator's Guide* for more information about automated maintenance tasks

Configuring the Automatic SQL Tuning Advisor

The following are some configuration tasks that you might want to perform for the Automatic SQL Tuning Advisor:

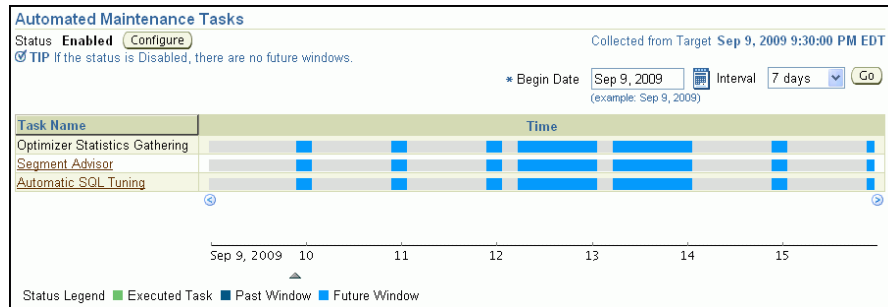
- Enable automatic implementation of SQL profile recommendations.

Automatic implementation is disabled by default.

- Select the maintenance windows (time periods) in which the advisor runs.
The Automatic SQL Tuning Advisor runs in all maintenance windows by default.
- Modify the start time and duration of existing maintenance windows or create new maintenance windows.

To configure the Automatic SQL Tuning Advisor:

1. Go to the Database Home page, logging in as user SYS.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. Click **Server** to display the Server page.
3. In the Oracle Scheduler section, click **Automated Maintenance Tasks**.
The Automated Maintenance Tasks page appears.



4. Click **Configure**.

The Automated Maintenance Tasks Configuration page appears.

The screenshot shows the 'Automated Maintenance Tasks Configuration' page. At the top, the 'Global Status' is 'Enabled'. Under 'Task Settings', three tasks are listed: 'Optimizer Statistics Gathering' (Enabled), 'Segment Advisor' (Enabled), and 'Automatic SQL Tuning' (Enabled). Each task has a 'Configure' button. Below this is the 'Maintenance Window Group Assignment' section, which includes an 'Edit Window Group' button and a table. The table has columns for 'Window', 'Optimizer Statistics Gathering', 'Segment Advisor', and 'Automatic SQL Tuning'. Each of the last three columns has 'Select All' and 'Select None' options. All tasks are assigned to all maintenance windows listed in the table.

Window	Optimizer Statistics Gathering	Segment Advisor	Automatic SQL Tuning
MID_MORNING_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MID_AFTERNOON_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MID_EVENING_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
WEDNESDAY_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
THURSDAY_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
FRIDAY_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SATURDAY_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SUNDAY_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MONDAY_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
TUESDAY_WINDOW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

5. (Optional) To disable the Automatic SQL Tuning Advisor, in the Task Settings section, click the **Disabled** option for Automatic SQL Tuning.

6. (Optional) To prevent the Automatic SQL Tuning Advisor from running in particular maintenance windows, in the Maintenance Window Group Assignment section, deselect check boxes under the Automatic SQL Tuning heading.

7. Click **Apply** to save any changes made so far.

A confirmation message is displayed.

8. To enable automatic implementation of SQL profile recommendations, complete the following steps:

- a. In the Task Settings section, next to the Automatic SQL Tuning options, click **Configure**.

The Automatic SQL Tuning Settings page appears.

- b. Next to Automatic Implementation of SQL Profiles, click the **Yes** option.

You must be logged in as user `SYS` to change this option.

- c. Click **Apply**.

A confirmation message is displayed.

- d. In the *locator links* at the top, left-hand side of the page, click **Automated Maintenance Tasks Configuration** to return to the Automated Maintenance Tasks Configuration page.

9. (Optional) To make changes to the start time and duration of existing maintenance windows, to disable individual maintenance windows, or to create additional maintenance windows, click **Edit Window Group**.

The Edit Window Group page appears. From this page you can change the settings of individual windows or you can add or remove windows to or from the window group `MAINTENANCE_WINDOW_GROUP`.

See the online Help for this page for more information.

Note: If you create a new window (time period) to run automated maintenance tasks, then you must add that window to `MAINTENANCE_WINDOW_GROUP`.

See Also:

- *Oracle Database Administrator's Guide* for more information about maintenance windows and how to configure them
- ["About the Automatic SQL Tuning Advisor"](#) on page 10-26

Viewing Automatic SQL Tuning Results

You can track the activities of the Automatic SQL Tuning Advisor with Database Control.

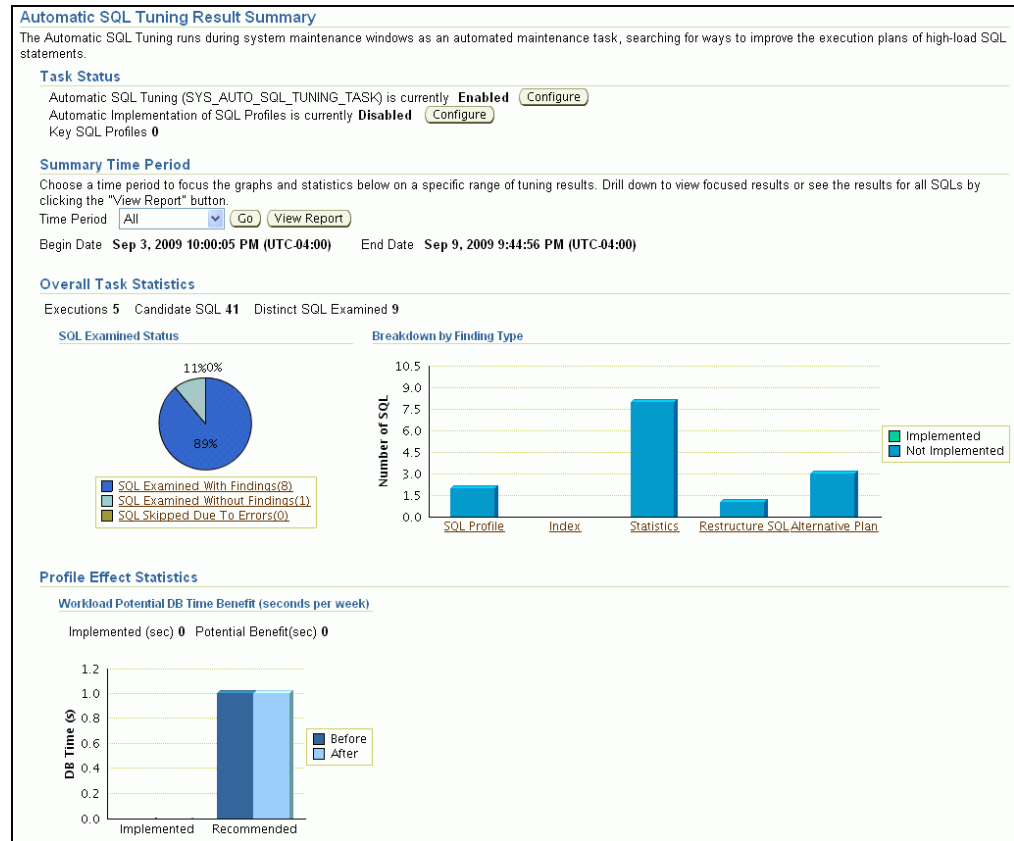
To view automatic SQL tuning results:

1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. Click **Server** to display the Server page.
3. In the Oracle Scheduler section, click **Automated Maintenance Tasks**.

The Automated Maintenance Tasks page appears.

4. Click **Automatic SQL Tuning**.

The Automatic SQL Tuning Result Summary page appears, showing graphical summaries of the Automatic SQL Tuning Advisor activities and findings.



5. To view recommendations, click **View Report** under the heading Task Activity Summary.

The Automatic SQL Tuning Result Details page appears, showing the SQL statements for which recommendations were made during the designated period.

Automatic SQL Tuning Result Details: All Analyzed SQLs							
Begin Date Jun 23, 2009 2:00:04 PM (UTC-07:00)				End Date Jun 24, 2009 11:03:54 AM (UTC-07:00)			
Recommendations							
Only profiles that significantly improve SQL performance were implemented.							
<input type="button" value="View Recommendations"/> <input type="button" value="Implement All SQL Profiles"/>							
Select	SQL Text	Parsing Schema	SQL ID	Weekly DB Time Benefit(sec)	Per-Execution % Benefit	Statistics	S Pr
<input checked="" type="radio"/>	SELECT COUNT(*) FROM HR.EMPLOYEES E, HR....	SH	f8ybh6brs1uq2	162.57	22		(22
<input type="radio"/>	SELECT :B1 TASK_ID, F.FINDING_ID FINDING...	DBSNMP	a8j39qb13tqkr	39.29	44	✓	(44
<input type="radio"/>	select unique MESSAGE_ID, SUBSYSTEM, LAN...	SYSMAN	9gj51ht8bu7g5	17.71	93	✓	
<input type="radio"/>	select unique MESSAGE_ID, SUBSYSTEM, LAN...	SYSMAN	49x4ct0cg6avj	14.99	96	✓	
<input type="radio"/>	select unique MESSAGE_ID, SUBSYSTEM, LAN...	SYSMAN	5x51kr74y26t2	14.50	75	✓	

A green check mark in the SQL Profile column indicates a recommendation that was automatically implemented.

By default, automatic implementation is disabled. See "[Configuring the Automatic SQL Tuning Advisor](#)" on page 10-27 for instructions for enabling it.

- (Optional) Select a SQL statement in the Recommendations table (based on the SQL Text column), and then click **View Recommendations**.

The Recommendations for SQL ID page appears, describing each recommendation for the statement in detail. On this page, you can select a recommendation and then click **Implement** to implement it.

See Also:

- Oracle Database 2 Day + Performance Tuning Guide
- ["About the Automatic SQL Tuning Advisor"](#) on page 10-26

Running the SQL Tuning Advisor

Use the SQL Tuning Advisor for tuning SQL statements. Typically, you run this advisor in response to an ADDM performance finding that recommends its use. You can also start the SQL Tuning Advisor manually. One reason is to tune statements that the Automatic SQL Tuning Advisor has not considered for tuning.

As described in "[About the SQL Advisors](#)" on page 10-25, the SQL Tuning Advisor can select SQL statements to tune from several sources. The following scenario assumes that you want to tune the SQL statements with the most activity:

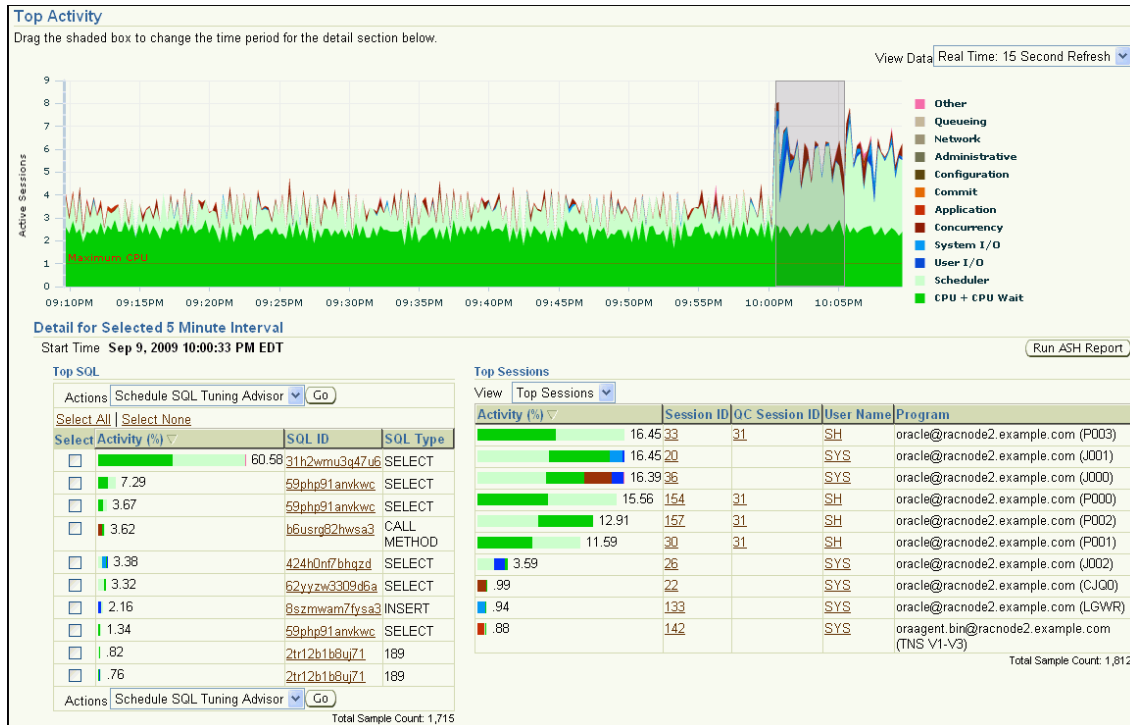
To run the SQL Tuning Advisor:

- Go to the Database Home page.
 - See "[Accessing the Database Home Page](#)" on page 3-6.
- At the bottom of the page, under Related Links, click **Advisor Central**.
- On the Advisor Central page, click **SQL Advisors**.
- On the SQL Advisors page, click **SQL Tuning Advisor**.

The Schedule SQL Tuning Advisor page appears.

- In the Overview section, select the **Top Activity** data source link.

The Top Activity page appears. This page has a timeline graph showing the database activity based on the active sessions over the last hour.



- (Optional) Expand the timeline by selecting **Historical** from the View Data list at the upper right-hand side of the page.
- Select an interval to analyze by clicking the bar under the timeline graph.
The Top SQL and Top Sessions tables show the activity within the selected period.
- In the Top SQL section, select one or more SQL statements. Make sure the **Actions** drop-down list shows Schedule SQL Tuning Advisor, and then click **Go**.
The Schedule SQL Tuning Advisor page returns, indicating the statements that are to be tuned.
- Enter a task name and description, select the scope for the analysis (**Comprehensive** or **Limited**), and select a start time for the analysis task.
- Click **Submit**.

A Processing SQL Tuning Advisor Task page appears. When the task is complete, the SQL Tuning Result Summary page appears, showing a summary of the SQL Tuning Advisor findings for the SQL statements analyzed.

To view the recommendations, click **Show all results**. The recommendations can include one or more of the following:

- Accept the generated SQL profile.
- Gather optimizer statistics on objects with stale or no statistics.
- Rewrite a query for better performance.

- Create an index to offer alternate, faster access paths to the query optimizer.
11. To view recommendations for a specified SQL statement, select a statement from the list of recommendations, and then click **View Recommendations**.

The Recommendations for SQL ID page appears, showing one or more recommendations for the statement.

12. (Optional) Select a recommendation, and then click **Implement**.

After you confirm that you want to implement a new profile, you are returned to the Recommendations for SQL ID page, and a confirmation notice is printed across the top of the page.

See Also:

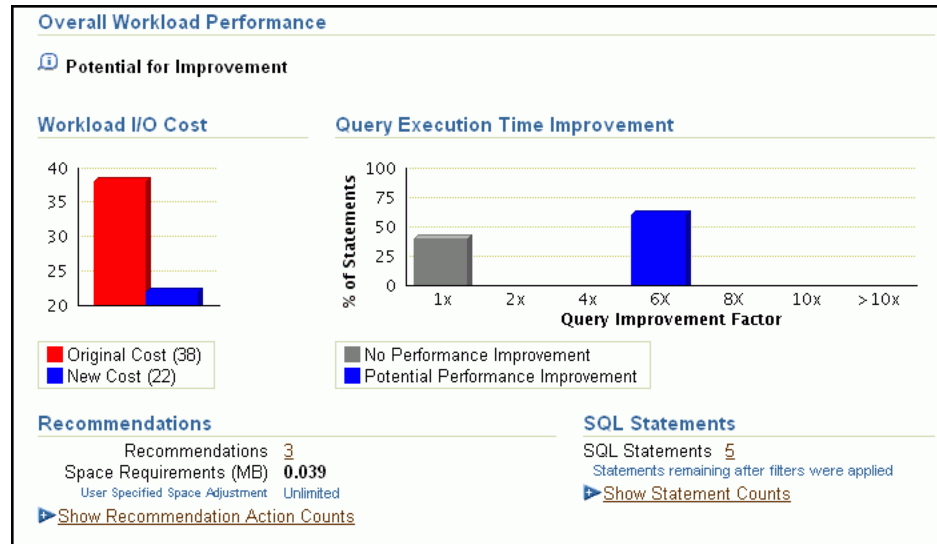
- ["About the SQL Advisors"](#) on page 10-25
- ["About the Automatic SQL Tuning Advisor"](#) on page 10-26

Running the SQL Access Advisor

You run the SQL Access Advisor to get recommendations for improving the performance of a workload. You can run it on a periodic basis to avoid performance problems, or run it to verify schema change recommendations from the SQL Tuning Advisor.

To run the SQL Access Advisor:

1. Go to the Database Home page.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. At the bottom of the page, under Related Links, click **Advisor Central**.
3. On the Advisor Central page, click **SQL Advisors**.
4. On the SQL Advisors page, click **SQL Access Advisor**.
A page appears prompting you for initial options. Click **Help** for information about the options for this page.
5. Select an option, and then click **Continue**.
6. The Workload Source page appears. It is the first page of the SQL Access Advisor wizard.
Click **Help** to obtain help for this and subsequent wizard pages. Follow directions and make the required selections for each wizard page, clicking **Next** when you are ready to proceed to the next page.
7. Continue through the wizard until you reach the Review page. Review your selections and then click **Submit** to start the analysis.
The Advisor Central page appears, displaying a confirmation message indicating that your task has been started successfully.
8. Click the **Refresh** button (not your browser's Refresh icon) to view the status of your task.
9. When your SQL Access Advisor task has completed, select **View Result**.
The Result for Task page appears.
The Summary subpage shows you the potential for improvement under the headings Workload I/O Cost and Query Execution Time Improvement.



The Recommendations subpage shows the recommendations, if any, for improving performance. A recommendation might consist of, for example, a SQL script with one or more CREATE INDEX statements, which you can run by clicking **Schedule Implementation**.

See Also:

- ["About the SQL Advisors"](#) on page 10-25
- *Oracle Database 2 Day + Performance Tuning Guide*

Optimizing Memory Usage with the Memory Advisors

This section contains:

- [About the Memory Advisors](#)
- [Example: Obtaining Memory Sizing Advice for ASMM](#)

See Also:

- ["Managing Memory"](#) on page 5-15
- ["About Advisors"](#) on page 10-23

About the Memory Advisors

Adequate physical memory has a significant impact on the performance of your Oracle Database. With its automatic memory management capabilities, Oracle Database can automatically adjust the memory distribution among the various SGA and PGA components for optimal performance. These adjustments are made within the boundaries of the total amount of memory that you allocate to the database.

ADDM periodically evaluates the performance of your database to determine performance problems. If ADDM finds that the current amount of available memory is inadequate and adversely affecting performance, then it can recommend that you increase memory allocations. You can select new memory allocations using the Memory Advisors.

Additionally, you can use the Memory Advisors to perform what-if analysis on the following:

- The database performance benefits of adding physical memory to your database
- The database performance impact of reducing the physical memory available to your database

With the Memory Advisors, you can obtain memory sizing advice as follows:

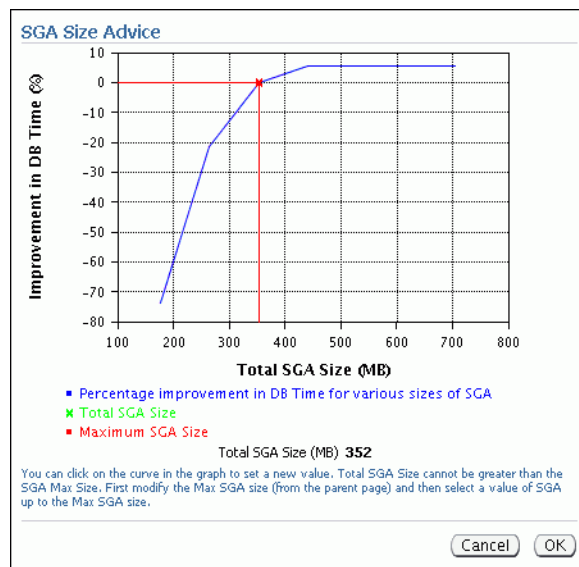
- If automatic memory management is enabled, then you can get advice for setting the target amount of memory to allocate to the Oracle instance.
- If automatic memory management is disabled and automatic shared memory management is enabled, then you can get advice on configuring the target sizes of the SGA and instance PGA.
- If only manual shared memory management is enabled, then you can get advice on sizing the shared pool, buffer cache, and instance PGA.

Example: Obtaining Memory Sizing Advice for ASMM

The following steps describe how to obtain memory sizing advice when automatic shared memory management (ASMM) is enabled.

To obtain memory sizing advice for ASMM:

1. Go to the Database Home page.
See "[Accessing the Database Home Page](#)" on page 3-6.
2. At the bottom of the page, click **Advisor Central**.
3. On the Advisor Central page, click **Memory Advisors**.
The Memory Advisors page appears.
4. On the SGA subpage, next to the Total SGA Size field, click **Advice**.
The SGA Size Advice child page appears in a separate window.



Improvement in DB Time (%) is plotted against Total SGA size. A higher number for Improvement in DB Time is better for performance. In this sample figure, the graph indicates that increasing the SGA size greater than 450 MB results in no performance gain. Thus, 450 MB is the recommended optimal SGA size.

Click **OK** to close the SGA Size Advice child page.

5. Near the top of the Memory Advisors page, click **PGA** to display the PGA subpage.
6. Next to Aggregate PGA Target, click **Advice**.

The PGA Target Advice page appears, plotting cache hit percentage against PGA target size.

The 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 automatically recommends adjusting this value in a performance finding.

Click **OK** to close the PGA Target Advice page.

See Also:

- ["Managing Memory"](#) on page 5-15

Monitoring and Tuning: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this section, and includes annotated screenshots.

To view the Monitoring OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/monitoring/monitoring.htm

Investigating, Reporting, and Resolving Problems

Oracle Enterprise Manager Database Control (Database Control) includes a feature called the Support Workbench that enables you to view and investigate critical database errors, report these errors to Oracle Support Services, and in some cases, resolve the errors.

This chapter explains critical errors and describes how to use the Database Home page and the Support Workbench to do the following:

- View critical error alerts
- View diagnostic data for critical errors
- Package diagnostic data for upload to Oracle Support Services
- Create and track a service request
- Repair some classes of critical errors

The following topics are covered:

- [About Database Problems and Incidents](#)
- [Investigating, Reporting, and Resolving a Problem](#)
- [Viewing Problems Using the Enterprise Manager Support Workbench](#)

About Database Problems and Incidents

To help diagnosis and resolution of critical errors, Oracle Database Release 11g introduces two concepts for Oracle Database: problems and incidents.

A **problem** is a critical error in the database. Critical errors manifest as internal errors, such as ORA-00600, and other severe errors, such as ORA-07445 (operating system exception) or ORA-04031 (out of memory in the shared pool portion). Problems are tracked in the **Automatic Diagnostic Repository (ADR)**. The ADR is a file-based repository for storing diagnostic data. Because this repository is stored outside the database, the diagnostic data is available even when the database is down. As of Release 11g, the alert log, all trace and dump files, and other diagnostic data are also stored in the ADR.

Each problem has a **problem key**, which is a text string that describes the problem. The problem key includes the error code (such as ORA 600), and in some cases, one or more error parameter values or other information.

An **incident** is a single occurrence of a problem. When a problem occurs multiple times, an incident is created for each occurrence. Incidents are timestamped and tracked in the Automatic Diagnostic Repository (ADR).

Each incident is identified by a numeric incident ID, which is unique within the ADR. When an incident occurs, the database performs the following steps:

- Makes an entry in the alert log.
- Sends an incident alert to Enterprise Manager.
- Gathers first-failure diagnostic data about the incident (such as trace files).
- Tags the diagnostic data with the incident ID.
- Stores the data in an ADR subdirectory created for that incident.

Each incident has a problem key and is mapped to a single problem.

Diagnosis and resolution of a critical error usually starts with an incident alert. The incident alert is displayed on the Enterprise Manager Database Home page. You can then view the problem and its associated incidents with Enterprise Manager.

See Also:

- ["Task 1 – View Critical Error Alerts in Enterprise Manager"](#) on page 11-4
- *Oracle Database Administrator's Guide* for more information about the ADR

Investigating, Reporting, and Resolving a Problem

This section describes the typical set of tasks that you perform to investigate and report a problem (critical error), and in some cases, resolve the problem. It includes the following topics:

- [Roadmap: Investigating, Reporting, and Resolving a Problem](#)
- [Task 1 – View Critical Error Alerts in Enterprise Manager](#)
- [Task 2 – View Problem Details](#)
- [Task 3 – \(Optional\) Create a Service Request](#)
- [Task 4 – Package and Upload Diagnostic Data to Oracle Support Services](#)
- [Task 5 – Track the Service Request and Implement Any Repairs](#)
- [Task 6 – Close Incidents](#)

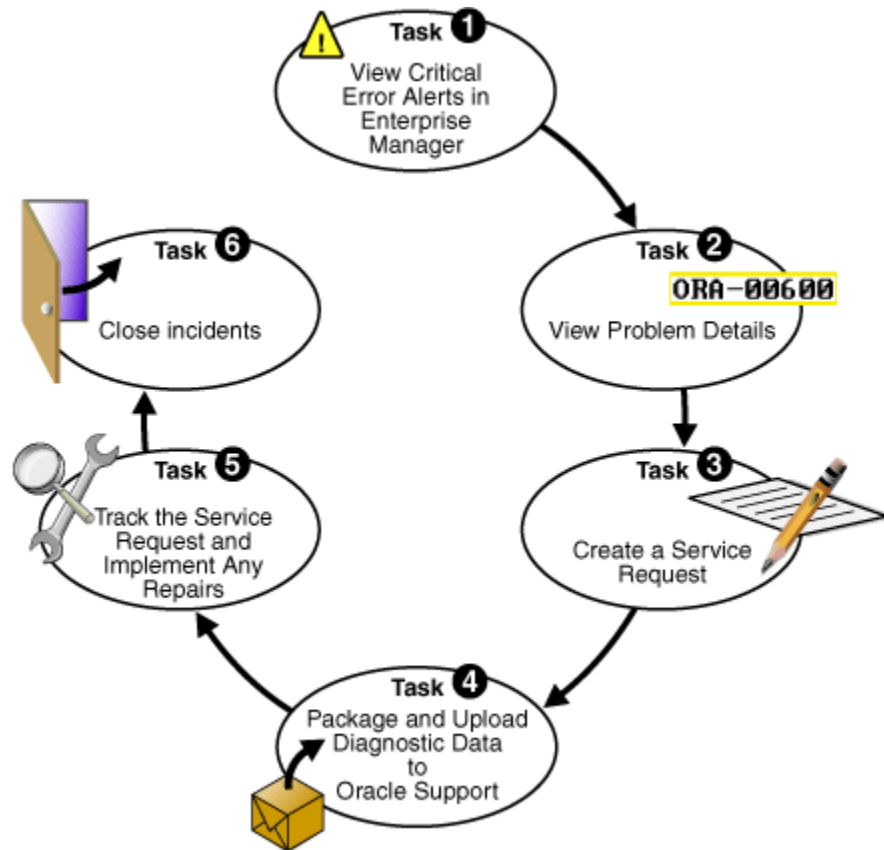
Note: The workflow described in this section includes only the minimum tasks that are required to investigate, report, and in some cases, repair a problem. See *Oracle Database Administrator's Guide* for a more complete workflow that includes additional diagnostics-gathering activities and data customization activities that you can do before uploading the diagnostic data to Oracle Support Services. In some cases, these additional activities may result in a shorter time to problem resolution.

Roadmap: Investigating, Reporting, and Resolving a Problem

You can begin investigating a problem by starting from the Support Workbench Home page. However, the more typical workflow begins with a critical error alert on the Database Home page. This documentation provides an overview of that workflow.

Figure 11-1 illustrates the basic tasks that you complete when encountering a problem.

Figure 11-1 Workflow for Investigating, Reporting, and Resolving a Problem



The following are task descriptions. Subsequent sections provide details for each task.

- **Task 1 – View Critical Error Alerts in Enterprise Manager** on page 11-4
Start by accessing the Database Home page in Enterprise Manager, and reviewing critical error alerts. Select an alert for which to view details. From the alert details page, go to the Problem Details page.
- **Task 2 – View Problem Details** on page 11-7
Examine the problem details and view a list of incidents that were recorded for the problem. Display findings from any health checks that were automatically run.
- **Task 3 – (Optional) Create a Service Request** on page 11-8
Create a service request using My Oracle Support and optionally record the service request number with the problem information. If you skip this step, then you can create a service request later, or the Support Workbench can create one for you.

- **Task 4 – Package and Upload Diagnostic Data to Oracle Support Services** on page 11-10

Invoke a guided workflow (a *wizard*) that automatically packages all gathered diagnostic data for a problem and uploads the data to Oracle Support Services.

- **Task 5 – Track the Service Request and Implement Any Repairs** on page 11-12

Optionally maintain an activity log for the service request in the Support Workbench. If appropriate, run Oracle advisors to help repair SQL failures or corrupted data.

- **Task 6 – Close Incidents** on page 11-14

Set the status for one, some, or all incidents for the problem to Closed.

See Also:

- "Viewing Problems Using the Enterprise Manager Support Workbench" on page 11-14
- "About Database Problems and Incidents" on page 11-1

Task 1 – View Critical Error Alerts in Enterprise Manager

You begin the process of investigating problems (critical errors) by reviewing critical error alerts on the Database Home page.

To view critical error alerts:

1. Go to the Database Home page in Enterprise Manager.
See "Accessing the Database Home Page" on page 3-6.
2. In the Alerts section, examine the table of alerts.

Critical error alerts are indicated by an X in the Severity column, and the text "Incident" in the Category column.

Note: You may have to click the hide/show arrowhead icon next to the Alerts heading to display the alerts table.

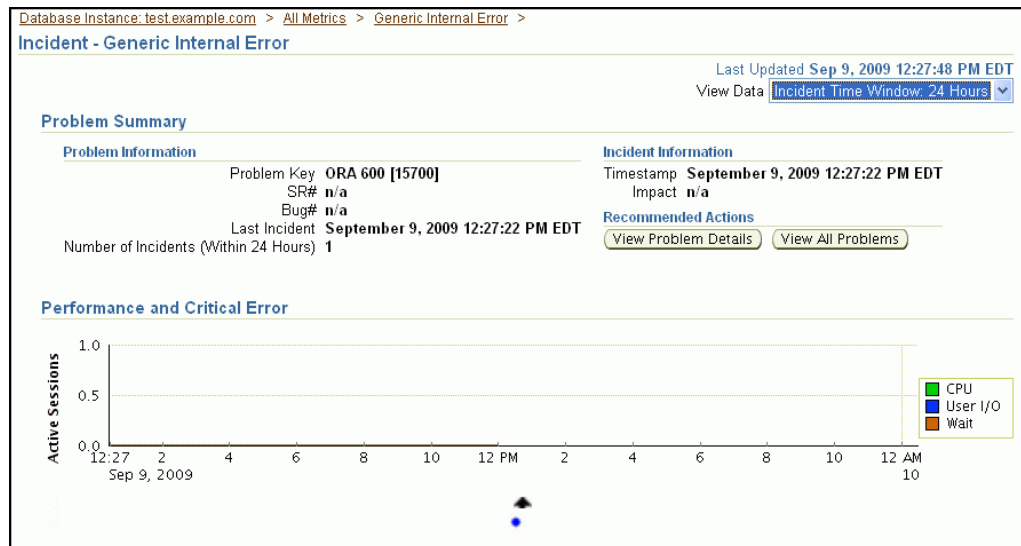
Severity	Category	Name	Impact	Message	Alert Triggered
✘	Incident	Generic Internal Error		Internal error (ORA-600(15700)) detected in /u01/app/oracle/diag/rdbms/test/test/alert/log.xml at time/line number: Wed Sep 9 12:27:22 2009/8777.	Sep 9, 2009 12:27:48 PM
✘	Incident Status	Generic Internal Error Status		Generic internal errors have been found in the alert log.	Sep 9, 2009 12:27:48 PM
⚠	User Audit	Audited User		User SYS logged on from racnode2.example.com.	Sep 3, 2009 3:49:42 PM
⚠	Tablespaces Full	Tablespace Space Used (%)		Tablespace TEST_MSSM_NEW is 86 percent full	Sep 8, 2009 12:18:11 PM

3. (Optional) In the Category list, select **Incident** to view alerts of type Incident only.
4. In the **Message** column, click the message of the critical error alert to investigate.

An Incident detail or Data Failure page appears for the type of incident you selected. For example, if you clicked a message about an ORA-600 error, then the Incident - Generic Internal Error page appears.

This page displays the following:

- Problem information, including the number of incidents for the problem
- A Performance and Critical Error graphical timeline for the 24-hour time period in which the critical error occurred
- Alert details, including severity, timestamp, and message
- Controls that enable you to clear the alert or record a comment about it



5. Review the Performance and Critical Error graphical timeline, and note any time correlation between performance issues and the critical error. Optionally clear the alert or leave a comment about it.
6. Perform one of the following actions:
 - To view the details of the problem associated with the critical error alert that you are investigating, proceed with [Task 2 – View Problem Details](#) on page 11-7.
 - If the graphical timeline shows a large number of different problems during the 24-hour time period and you want to view a summary of all those problems, then complete these steps:
 - Click **View All Problems**.
 - The Support Workbench Home page appears.
 - View problems and incidents as described in ["Viewing Problems Using the Enterprise Manager Support Workbench"](#) on page 11-14.
 - Select a single problem and view problem details, as described in ["Viewing Problems Using the Enterprise Manager Support Workbench"](#) on page 11-14.
 - Continue with [Task 3 – \(Optional\) Create a Service Request](#) on page 11-8.

Task 2 – View Problem Details

You continue your investigation with the Problem Details page.

To view problem details:

1. On the Incident detail or Data Failure page, click **View Problem Details**.

The Problem Details page appears, showing the Incidents subpage.

2. (Optional) To view incident details, in the Incidents subpage, select an incident, and then click **View**.

The Incident Details page appears, showing the Dump Files subpage.

3. (Optional) On the Incident Details page, select **Checker Findings** to view the Checker Findings subpage.

This page displays findings from any health checks that were automatically run when the critical error was detected.

See Also:

- *Oracle Database Administrator's Guide* for information about health checks and checker findings

Task 3 – (Optional) Create a Service Request

At this point, you can create an Oracle Support service request and record the service request number with the problem information. If you choose to skip this task, then the Support Workbench automatically creates a draft service request for you in Task 4.

To create a service request:

1. On the Problem Details page, in the Investigate and Resolve section, click **Go to My Oracle Support**.

Tip: If you followed the steps described in Task 2, then from the Incident Details page, click Problem Details in the breadcrumb to return to the Problem Details page.

Database Instance: test.example.com > Support Workbench > Logged in As SYS

Problem Details: ORA 600 [15700] Page Refreshed September 9, 2009 12:59:00 PM EDT [Refresh](#)

Summary

SR# .. [Edit](#)

Bug# .. [Edit](#)

Active **Yes**

Packaged **No**

Number of Incidents **2**

First Incident [September 9, 2009 12:27:22 PM EDT](#)

Last Dumped Incident

Timestamp [September 9, 2009 12:49:42 PM EDT](#)

Incident Source **System Generated**

Impact

Checkers Run **0**

Checker Findings **0**

Investigate and Resolve

[Go to My Oracle Support](#) [Quick Package](#)

[Self Service](#) [Oracle Support](#)

Assess Damage

[Run Checkers](#)

[Database Instance Health](#)

Diagnose

[Alert Log](#)

[Related Problems Across Topology](#)

[Diagnostics for Last Dumped Incident](#)

[Go to My Oracle Support and Research](#)

Resolve

[SQL Repair Advisor](#) (Recommended)

Incidents [Activity Log](#)

Status [Open Incidents](#) Data Dumped [Yes](#) [Go](#)

[View](#) [Close](#)

[Select All](#) | [Select None](#) | [Show All Details](#) | [Hide All Details](#)

Select Details	ID	Description	Data Dumped	Active	Status	Timestamp
<input type="checkbox"/> Show	7433	ORA-600 [15700] [1] [1]	Yes	Yes	Ready	September 9, 2009 12:49:42 PM EDT
<input type="checkbox"/> Show	7417	ORA-600 [15700] [1] [1]	Yes	Yes	Ready	September 9, 2009 12:27:22 PM EDT

The My Oracle Support Login and Registration page appears in a new browser window.

2. Log in to My Oracle Support and create a service request in the usual manner. (Optional) Remember the service request number (SR#) for the next step.
3. (Optional) Return to the Problem Details page, and then do the following:
 - a. In the Summary section, click the **Edit** button that is adjacent to the SR# label.
 - b. In the page that opens, enter the SR#, and then click **OK**.

The SR# is recorded in the Problem Details page. This is for your reference only.

Task 4 – Package and Upload Diagnostic Data to Oracle Support Services

For this task, you use the quick packaging process of the Support Workbench to package and upload the diagnostic information for the problem to Oracle Support Services. Quick packaging has a minimum of steps, organized in a guided workflow (a wizard). The wizard assists you with creating an incident package (referred to as a *package*) for a single problem, creating a ZIP file from the package, and uploading the file. With quick packaging, you are not able to edit or otherwise customize the diagnostic information that is uploaded. Using quick packaging is the direct, straightforward method to package and upload diagnostic data.

To edit or remove sensitive data from the diagnostic information, enclose additional user files (such as application configuration files or scripts), or perform other customizations before uploading, you must use the custom packaging process. See *Oracle Database Administrator's Guide* for instructions. When you complete those instructions, you may continue with [Task 5 – Track the Service Request and Implement Any Repairs](#) on page 11-12.

Note: The Support Workbench uses Oracle Configuration Manager to upload the diagnostic data. If Oracle Configuration Manager is not installed or properly configured, then the upload may fail. In this case, a message is displayed with a request that you upload the file to Oracle Support manually. You can upload files manually with My Oracle Support.

For more information about Oracle Configuration Manager, see *Oracle Configuration Manager Installation and Administration Guide*.

To package and upload diagnostic data to Oracle Support:

1. On the Problem Details page, in the Investigate and Resolve section, click **Quick Package**.

The Create New Package page of the Quick Packaging wizard appears.

2. (Optional) Enter a package name and description.
3. Fill in the remaining fields on the page. If you have created a service request for this problem, then select the **No** option button for Create new Service Request (SR).

If you select the Yes option button for Create new Service Request (SR), then the Quick Packaging wizard creates a draft service request on your behalf. You must later log in to My Oracle Support and fill in the details of the service request.

Click **Next**.

The Quick Packaging wizard displays a page indicating that it is processing the command to create a new package. When it finished, the Quick Packaging: View Contents page is displayed.

4. Review the contents on the page, making note of the size of the created package, then click **Next**.

The Quick Packaging: View Manifest page appears.

5. Review the information on this page, making note of the location of the manifest (listed next to the heading Path). After you have reviewed the information, click **Next**.

The Quick Packaging: Schedule page appears.

6. Choose either **Immediately**, or **Later**. If you select **Later**, then you provide additional information about the time the package should be submitted to My Oracle Support. After you have made your choice and provided any necessary information, click **Submit**.

The Processing: Packaging and Sending the Package progress page appears.

When the Quick Packaging wizard is complete, the package that it creates remains available in the Support Workbench. You can modify the incident package with custom packaging operations (such as adding new incidents) and reupload the package at a later time.

Task 5 – Track the Service Request and Implement Any Repairs

After uploading diagnostic information to Oracle Support Services, you might perform various activities to track the service request, collect additional diagnostic information, and implement repairs. Among these activities are the following:

- Adding an Oracle bug number to the problem information.

To do so, on the Problem Details page, click the **Edit** button that is adjacent to the Bug# label. This is for your reference only.

- Adding comments to the problem activity log.

You may want to share problem status or history information with other DBAs in your organization. For example, you could record the results of your conversations with Oracle Support.

To add comments to the problem activity log:

1. Go to the Problem Details page for the problem, as described in "[Viewing Problems Using the Enterprise Manager Support Workbench](#)" on page 11-14.
2. Click **Activity Log** to display the Activity Log subpage.

Incidents		Activity Log	
User	Action	Description	Timestamp
SYS	Comment	Upload to Oracle Support failed because OCM was not configured. Ran the \$ORACLE_HOME/ccr/bin/setupCCR script to configure it.	September 9, 2009 1:54:07 PM EDT
SYS	Package	Sent upload file : packageId = 1 file = /u01/app/oracle/product/11.2.0/dbhome_2/racnode2_test/sysman/emd/state/ORA600157_20090909131503_COM_1.zip	September 9, 2009 1:36:04 PM EDT
SYS	Package	Created physical file : packageId = 1 file = /u01/app/oracle/product/11.2.0/dbhome_2/racnode2_test/sysman/emd/state/ORA600157_20090909131503_COM_1.zip	September 9, 2009 1:35:50 PM EDT
SYS	Comment	Created package : Id = 1 Name = ORA600157_20090909131503	September 9, 2009 1:25:55 PM EDT

3. In the **Comment** field, enter a comment, and then click **Add Comment**.

Your comment is recorded in the activity log.

- Adding new incidents, if they occur, to the package and reuploading the package to Oracle Support Services.

For this activity, you must use the custom packaging method described in *Oracle Database Administrator's Guide*.

- Running health checks.

See *Oracle Database Administrator's Guide* for information about health checks.

- Running a suggested Oracle advisor to implement repairs.

You can access the suggested advisor in one of the following ways:

- **Problem Details page**—In the Self-Service tab of the Investigate and Resolve section
- **Support Workbench Home page**—On the Checker Findings subpage
- **Incident Details page**—On the Checker Findings subpage

[Table 11-1](#) lists the advisors that help repair critical errors.

Table 11-1 Oracle Advisors that Help Repair Critical Errors

Advisor	Critical Errors Addressed	See
Data Recovery Advisor	Corrupted blocks, corrupted or missing files, and other data failures	"Performing Oracle Advised Recovery" on page 9-28
SQL Repair Advisor	SQL statement failures	<i>Oracle Database Administrator's Guide</i>

See Also:

- ["Viewing Problems Using the Enterprise Manager Support Workbench"](#) on page 11-14

Task 6 – Close Incidents

When a particular incident is no longer of interest, you can close it. By default, closed incidents are not displayed on the Problem Details page.

All incidents, whether closed or not, are purged after 30 days. You can disable purging for an incident on the Incident Details page.

To close incidents:

1. Go to the Support Workbench Home page.
See ["Viewing Problems Using the Enterprise Manager Support Workbench"](#) on page 11-14.
2. Select the desired problem, and then click **View**.
The Problem Details page appears.
3. Select the incidents to close and then click **Close**.
A confirmation page appears.
4. Click **Yes** to close the incident.

Viewing Problems Using the Enterprise Manager Support Workbench

You use the Enterprise Manager Support Workbench home page to view all problems or only those within a specified time period.

To access the Support Workbench home page:

1. Go to the Database Home page in Enterprise Manager.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. Click **Software and Support** to view the Software and Support page.
3. In the Support section, click **Support Workbench**.

The Support Workbench home page appears, showing the Problems subpage. By default the problems from the last 24 hours are displayed.

Database Instance: test.example.com > Logged in As SYS
 Support Workbench
 Page Refreshed September 9, 2009 2:02:50 PM EDT Refresh

Problems (1) Checker Findings (0) Packages (1)

New Problems in Last 24 Hours 1 All Active Problems 1 All Problems 1
 New Incidents in Last 24 Hours 2 All Active Incidents 2 All Incidents 2

View Last 24 Hours Search Go Advanced Search

View Package

Select All | Select None | Show All Details | Hide All Details

Select Details	ID	Description	Number Of Incidents	Last Incident	Last Comment	Active	Packaged	SR#
<input type="checkbox"/>	7433	ORA-600 [15700]	2	September 9, 2009 12:49:42 PM EDT	Closed Incident: 7433	Yes	Yes	

Incidents (2)

7433	ORA-600 [15700] [1] [1]	September 9, 2009 12:49:42 PM EDT
7417	ORA-600 [15700] [1] [1]	September 9, 2009 12:27:22 PM EDT

- To view all problems, select **All** from the View list.
- (Optional) If the Performance and Critical Error section is hidden, then click the **Show/Hide** icon adjacent to the section heading to show the section.
 This section enables you to view any correlation between performance changes for your database and incident occurrences.
- (Optional) Under the Details column, click **Show** to display a list of all incidents for a problem, and then click an incident ID to display the Incident Details page.

To view details for a particular problem:

- On the Support Workbench home page, select the problem, and then click **View**.
 The Problem Details page appears, showing the Incidents subpage.
- (Optional) To view closed incidents and open incidents, select **All** from the View list.
- (Optional) To view details for an incident, select the incident, and then click **View**.
 The Incident Details page appears.
- (Optional) To view the contents of an associated trace file, on the Dump Files subpage of the Incident Details page, click the icon in the View Contents column for the trace file.
 Enter the host credentials, then click **Continue**. The contents of the trace file are displayed. Click **OK** to return.
- (Optional) To view checker findings for the incident, on the Incident Details page, click **Checker Findings**.
 The Checker Findings subpage appears.
- (Optional) On the Incident Details page, to view the user actions that are available to you for the incident, click **Additional Diagnostics**. Each user action provides a way for you to gather additional diagnostics for the incident or its problem.

Managing Oracle Database Software

This chapter describes how to keep your Oracle Database software up-to-date. It contains the following sections:

- [About Software Management and Patch Releases](#)
- [Upgrading a Database](#)
- [Removing Oracle Database Software](#)
- [Managing Oracle Software: Oracle By Example Series](#)

About Software Management and Patch Releases

Software management involves keeping your Oracle Database software up-to-date with the latest product fixes. When a product defect, or a **bug**, is discovered, a patch is created to fix the problem. A **patch** corrects a single defect in the installed software. Individual patches, also referred to as **interim patches**, are made available to customers who for business reasons cannot wait until the next patch set to receive the product fix.

Oracle periodically issues maintenance releases for its software, in the form of patch sets. A **patch set** is a collection of product fixes that have been released up to the time of the maintenance release. Patch sets are fully tested and integrated product fixes. All the product fixes in the patch set have been tested and are certified to work with each other.

Every patch or patch set is associated with a bug number for identification purposes. Patch sets are also associated with version numbers. For example, if you use Oracle Database 11g Release 11.1.0.6, then an available patch set might be 11.1.0.7. The version number of the patched software does not change if an interim patch is applied.

Every patch has an associated README file that describes how it fixes the software. The README file also has instructions for applying the patch.

Every patch set is accompanied by a Patch Set Notes file that contains installation instructions and information about the product fixes contained within the patch set. When you apply a patch set to your Oracle software, you change the maintenance release number for your installed software. Applying a patch set affects the software residing in your Oracle home only, with no change to the data in the database.

You can use Oracle OPatch and Oracle Universal Installer to apply patches and patch sets respectively. Alternatively, you can automate the application of these patches using Oracle Enterprise Manager Grid Control. Use of these products is beyond the scope of this documentation.

See Also: *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX*

Upgrading a Database

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

This section contains these topics:

- [Overview of Database Upgrade Assistant](#)
- [Database Versions Supported by DBUA](#)
- [Starting DBUA](#)
- [Upgrading a Database Using DBUA](#)

Overview of Database Upgrade Assistant

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

DBUA can be used to upgrade databases created using any edition of the Oracle Database software, including Express Edition (XE) databases.

Pre-Upgrade Checks

DBUA 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 data types 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, DBUA automatically performs the following tasks:

- Modifies or creates new required tablespaces
- Invokes the appropriate upgrade scripts
- Archives the online redo log files
- Disables archiving during the upgrade phase (to improve performance)

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

Support for Oracle Real Application Clusters

DBUA is fully compliant with Oracle Real Application Clusters (Oracle RAC) environments. In Oracle RAC environments, DBUA upgrades all database and configuration files on all nodes in the cluster.

About Upgrading Oracle Automatic Storage Management

Oracle ASM Configuration Assistant (ASMCA) enables you to upgrade an existing Oracle ASM instance to the current software level. However, the recommended practice is to upgrade an Oracle ASM instance with Oracle Universal Installer (OUI). OUI automatically defaults to upgrade mode when it detects an Oracle ASM instance at a previous release level.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for information about upgrading Oracle ASM

Support for Silent Mode

DBUA 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.

Database Versions Supported by DBUA

DBUA supports the following versions of Oracle Database for upgrading to Oracle Database 11g Release 2 (11.2):

- Oracle9i Release 2 (9.2.0.6) and beyond
- Oracle Database 10g release 1 (10.1)
- Oracle Database 10g release 2 (10.2)
- Oracle Database 11g release 1 (11.1)

If your database version is not in this list, then you must upgrade first to the closest release listed. You can then upgrade the database to Oracle Database 11g release 2 (11.2).

Starting DBUA

If you install the Oracle Database software only and specify that you are upgrading an existing database to the new Oracle Database release, then DBUA is launched automatically after the software installation. You can then continue as described in "[Upgrading a Database Using DBUA](#)" on page 12-5.

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

Be aware of the following before you begin using DBUA:

- You must run Net Configuration Assistant (NETCA) before running DBUA.
- It is not possible to upgrade a database with DBUA when the source and target Oracle homes are owned by different users.
- If the database instance is not running, then DBUA tries to start the instance with the default initialization parameter file. If that fails, then you are prompted to provide the name of the correct initialization parameter file or to start the instance. If the instance is up and running, then DBUA connects to it.

- If you stop the upgrade, but do not restore the database, then you should not restart DBUA until you start the database instance in `UPGRADE` mode using the Oracle Database 11g release 2 (11.2) software. You cannot go back to the original software version unless you restore your database.
- If you restore your database manually (not using DBUA), then remove the following file from the Oracle Database 11g release 2 (11.2) home directory before starting DBUA:

```
$ Oracle_home/cfgtoollogs/dbua/logs/Welcome_SID.txt
```

The presence of this file indicates to DBUA that this operation is a reattempt of a previous operation.

To start DBUA on Microsoft Windows:

1. Configure the operating system environment variables, as described in "[Configuring the Operating System Environment Variables](#)" on page 3-4.
2. Click **Start**, then select **Programs** (or **All Programs**)
3. Select **Oracle - HOME_NAME**
4. Select **Configuration and Migration Tools**
5. Select Database Upgrade Assistant

The Database Upgrade Assistant: Welcome window appears.

To start DBUA on any supported platform:

1. Open a command window.
2. Configure the operating system environment variables, as described in "[Configuring the Operating System Environment Variables](#)" on page 3-4.
3. Enter the following command:

```
dbua
```

The Database Upgrade Assistant: Welcome window appears.

Note: The `dbua` executable is typically found in your `Oracle_home/bin` directory.

See Also:

- *Oracle Database Upgrade Guide* for more information about Database Upgrade Assistant

Upgrading a Database Using DBUA

Complete the following steps to upgrade a database using DBUA. If you need help at any window or want to consult more documentation about DBUA, then click the **Help** button to access the online Help.

To upgrade a database using DBUA:

1. Start DBUA. See "[Starting DBUA](#)" on page 12-4.
2. At the Welcome window of DBUA, make sure the database being upgraded meets the specified conditions. Then, click **Next**.

3. At the Databases window, 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 run 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.

4. DBUA displays a message saying it is getting database information. DBUA analyzes the selected database, performing pre-upgrade checks and displaying warnings as necessary:
 - It checks for any online redo log files of a size less than 4 megabytes (MB). If such files are found, then 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.

If no problems are found, then the Diagnostic Destination window appears.

5. In the **Diagnostic Destination** field, do one of the following:
 - Leave the setting at its default value, which is the Oracle base directory.
 - Enter a new destination.
 - Click **Browse** and select a new destination.

Diagnostic Destination is the default location to store Oracle trace and diagnostic files. It replaces the initialization parameter settings for background dump destination, user dump destination and core dump destination from earlier Oracle Database releases.

Click **Next**.

The Move Database Files window appears.

6. Do one of the following:
 - Select **Do Not Move Database Files as Part of Upgrade**.
 - Select **Move Database Files During Upgrade**.

If you choose to move database files, then you must also select either File System or Oracle Automatic Storage Management (Oracle ASM).

Click **Next**.

The Recompile Invalid Objects window appears.

7. (Optional) Select **Recompile invalid objects at the end of upgrade** and modify the value of degree of parallelism.

When you upgrade your database to the new Oracle Database release, many of the PL/SQL modules in the database become invalid. By default, Oracle Database recompiles 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**. All the invalid PL/SQL modules are recompiled immediately after the upgrade is performed. The task of recompiling all the invalid PL/SQL modules in your database can take a significant amount of time and increase the time it takes to complete your database upgrade.

If you have multiple CPUs, then DBUA automatically adds a **Degree of Parallelism** menu to the Recompile Invalid Objects window. Parallel processing reduces the time it takes to recompile all the invalid PL/SQL modules in your

database. 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/utl1rp.sql` script, which is used to recompile stored PL/SQL and Java code.

Click **Next**.

The Backup window appears.

8. Select one of the following options:

- **I have already backed up my database.**
- **I would like this tool to back up the database.**

If you use DBUA to back up your database, then DBUA makes a copy of all database files in the directory that you specify in the **Backup Directory** field. 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.

In the **Backup Directory** field, do one of the following:

- Leave the setting at its default value
- Enter a different valid file system path. You cannot specify a raw device for the backup files.
- Click **Browse** and select a new backup destination.

Oracle strongly recommends that you back up your database before starting the upgrade. If errors occur during the upgrade, then you may need to restore the database from the backup.

In addition to creating a backup of your database, DBUA creates an executable script in the directory specified in the Backup Directory field. If needed, you can use the executable script appropriate to your system to restore the database files:

- Microsoft Windows: `db_name_restore.bat`
- Linux or UNIX systems: `db_name_restore.sh`

Click **Next**.

The Recovery Configuration window appears.

9. In the **Fast Recovery Area** field, do one of the following:

- Leave the setting at its default value
- Enter a different fast recovery area
- Click **Browse** and select a different fast recovery area.

The fast recovery area can be used to recover data that would otherwise be lost during a failure. This location is also used by Enterprise Manager if you have enabled local management and daily backups on the Database Control Management Options page.

In the **Fast Recovery Area Size** field, select the units you want from the list and do one of the following:

- Leave the setting at its default value.
- Enter a different value for Fast Recovery Area Size.
- Use the up and down arrows to set a different Fast Recovery Area Size.

In the Recovery Configuration window, you specify a fast recovery area and enable archiving. It is important to configure these features for your database so you can recover your data if a failure occurs.

Click **Next**.

The Summary window appears.

10. Review the list the initialization parameters that will be set for the database during the upgrade. Click **Back** to correct any errors, or click **Finish**.

Note: The database is not be available for general use during the upgrade process that begins when you click **Finish**.

A Progress window appears and DBUA begins to perform the upgrade.

11. You might encounter error messages with Ignore and Abort choices:
 - **Ignore** - Ignores the error, skips the current step, 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 DBUA, and complete the skipped steps.
 - **Abort** - Stops the upgrade process. DBUA prompts you to restore the database if the database backup was performed by DBUA. After the database has been restored, correct the error and restart DBUA to perform the upgrade again. If you do not restore the database, then 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 is complete. Click "OK" to see the results of the upgrade.

Click **OK**.

The Upgrade Results window appears.

12. Examine the results of the upgrade. The upgrade results summary describes the original and upgraded databases and changes made to the initialization parameters.

The upgrade results also include an Upgrade Details section that describes the steps performed during the database upgrade. This section 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 Upgrade Details section 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.

13. Click **Configure Database Passwords**.

The Password Management dialog box appears.

14. The Password Management dialog box enables you to change the default password for a user after you upgrade the database.

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

Click the **Lock Account?** column for a user to lock or unlock an account. A check mark indicates that the account is locked.

Enter a new password for a user in its **New Password** column.

Confirm the new password by entering it in the **Confirm Password** column.

Click **OK** to return to the Upgrade Results window.

15. Click **Restore** if you are not satisfied with the upgrade results.

Depending on the method you used to back up your database, the restore operation performs one of the following tasks:

- If you used DBUA to back up your database, then clicking Restore copies the original database files and the original database settings from the backup.
- If you used your own backup procedure to back up the database, then clicking Restore copies only the original database settings. To restore the database itself, copy the data files from the backup you created using your own backup utilities.

Click **Close** to quit DBUA if you are satisfied with the upgrade results.

DBUA removes the entry of the upgraded database from the old `listener.ora` file and restarts the listener.

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

See Also:

- *Oracle Database Upgrade Guide* for information about additional tasks that should be completed after upgrading a database

Removing Oracle Database Software

If you want to remove an Oracle software installation, you can use the Deinstallation tool to completely uninstall the software from your computer.

The Deinstallation Tool (`deinstall`) is available in Oracle home directories after installation as `$ORACLE_HOME/deinstall/deinstall`. The Deinstallation Tool is also available for download as a separate archive file from the Oracle Technology Network (OTN) site (<http://www.oracle.com/technology/software/products/database/index.html>).

To remove an Oracle home from your computer:

1. Execute the Deinstallation tool. The command you enter depends on whether you are executing the tool from an installed Oracle home or executing the tool after downloading it from OTN. Do one of the following:

If you are executing the tool from an installed Oracle home, the following is an example of the commands you would enter:

```
$ cd /u01/app/oracle
$ product/11.2.0/dbhome_1/deinstall/deinstall
```

If you are executing the tool after downloading and unzipping the archive file from OTN, enter the following command (replacing *Oracle_home* with the full path name for your Oracle home):

```
$ deinstall -home Oracle_home
```

Note: Do not shut down the Oracle database or stop any database processes prior to running the Deinstallation tool.

2. Enter the specified information at the prompts, or just press Enter to accept the default values.
3. When prompted, follow the directions to run the root script. You must be the root user to do so.

See Also:

- *Oracle Database Installation Guide* for your platform for more information about the Deinstallation tool

Managing Oracle Software: Oracle By Example Series

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

To view the Managing Oracle Software OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/managing/managing.htm

Administering Oracle Automatic Storage Management

This appendix discusses using Oracle Automatic Storage Management (Oracle ASM). Oracle ASM is Oracle's recommended storage management solution that provides an alternative to conventional volume managers, file systems, and raw devices. This chapter includes the following topics:

- [About Oracle Automatic Storage Management](#)
- [Overview of Disks, Disk Groups, and Failure Groups](#)
- [Installing Oracle Automatic Storage Management](#)
- [Accessing the Oracle Automatic Storage Management Home Page](#)
- [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 Oracle ASM-Managed Files](#)
- [Oracle Automatic Storage Management: Oracle By Example Series](#)

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for more information about Oracle Automatic Storage Management

About Oracle Automatic Storage Management

Oracle Automatic Storage Management (Oracle ASM) is an integrated, high-performance database file system and disk manager. You use Oracle ASM instead of an operating system file system to store your Oracle Database files and, optionally, other files.

Oracle ASM is based on the principle that the database should manage storage instead of requiring an administrator to do it. Oracle ASM eliminates the need for you to manage potentially thousands of database files.

Oracle ASM groups the disks in your storage system into one or more **disk groups**. You manage a small set of disk groups, and Oracle ASM automates the placement of the database files within those disk groups.

Oracle ASM provides the following benefits:

- **Striping**—Oracle 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, then pieces of each Oracle ASM file are written to all six disks. These pieces are known as **extents**. In this scenario, these pieces come in 1 MB chunks. 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 increases evenly. When reading the file, file extents are read from all six disks in parallel, greatly increasing performance.

- **Mirroring**—Oracle ASM can increase availability by optionally mirroring any file. Oracle 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, then Oracle ASM can continue to access affected files by accessing mirrored copies on the surviving disks in the disk group.

Oracle 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 reconsideration and dynamic rebalancing**—Oracle ASM permits you to add or remove disks from your disk storage system while the database is operating. When you add a disk, Oracle 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, Oracle ASM first rebalances the disk group by evenly relocating all file extents from the disk being removed to the other disks in the disk group.
- **Managed file creation and deletion**—Oracle ASM further reduces administrative tasks by enabling files stored in Oracle ASM disk groups to be **Oracle Managed Files**. Oracle ASM automatically assigns file names 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 Oracle ASM for your database file storage, instead of raw devices or the operating system file system. However, databases can have a mixture of Oracle ASM files and non-Oracle ASM files. Oracle Enterprise Manager includes a wizard that enables you to migrate non-Oracle ASM database files to Oracle ASM.

The Oracle ASM Instance

Oracle ASM is implemented as a special kind of Oracle instance, with its own System Global Area and background processes. The Oracle ASM instance is tightly integrated with the database instance. Every server running one or more database instances that use Oracle ASM for storage has an Oracle ASM instance. In an Oracle Real Application Clusters environment, there is one Oracle ASM instance for each node, and the Oracle ASM instances communicate with each other on a peer-to-peer basis. Only one Oracle ASM instance is required for each node, regardless of the number of database instances on the node.

Administering Oracle ASM

You administer Oracle ASM with Oracle Enterprise Manager Database Control (Database Control). To administer the Oracle ASM instance and Oracle ASM disk

groups, you must connect to the Oracle ASM instance as a user who has been granted the `SYSASM` system privilege. You can also administer Oracle ASM with Oracle Automatic Storage Management Command Line utility (ASMCMD) and SQL*Plus statements.

When you create the Oracle ASM instance either during installation or using Oracle Automatic Storage Management Configuration Assistant (ASMCA), the `SYSASM` is granted to the specified Oracle ASM administrator user.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for additional information about Oracle ASM
- ["Accessing the Oracle Automatic Storage Management Home Page"](#) on page A-4

Overview of Disks, Disk Groups, and Failure Groups

You configure Oracle ASM by creating disk groups that become the default location for files created in the database. The disk group type determines how Oracle 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 Oracle ASM). You use an external redundancy disk group if your storage system 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 as a unit. These disks are referred to as **Oracle ASM disks**. An Oracle ASM disk can be a disk device, a partition, or a network-attached file.

When an Oracle ASM instance starts, it automatically *discovers* all available Oracle ASM disks. **Discovery** is the process of finding all disks that were prepared for Oracle 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. Oracle 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 Oracle ASM disks that share a common potential failure mechanism. An example of a failure group is a set of small computer system interface (SCSI) disks sharing the same SCSI controller. Failure groups are used to determine which Oracle ASM disks to use for storing redundant copies of data. For example, if 2-way mirroring is specified for a file, then Oracle ASM automatically stores redundant copies of file extents in separate failure groups. Failure groups apply only to normal and high redundancy disk groups. You can define the failure groups in a disk group when you create or alter the disk group, or let Oracle ASM automatically place each disk into its own failure group. For most systems, the default failure groups work well to prevent data loss.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for additional information about failure groups, and for instructions for configuring failure groups

Installing Oracle Automatic Storage Management

This documentation assumes that Oracle Automatic Storage Management (Oracle ASM) is installed and configured on your single-instance server or on your Oracle Real Application Clusters (Oracle RAC) nodes. For information about installing and initially configuring Oracle ASM, for either a single-instance database or an Oracle RAC database, see *Oracle Grid Infrastructure Installation Guide for Linux*.

See Also:

- ["About Oracle Automatic Storage Management"](#) on page A-1

Accessing the Oracle Automatic Storage Management Home Page

All Oracle Automatic Storage Management (Oracle ASM) administration tasks begin with the Oracle ASM Home page in Database Control. The Oracle ASM Home page displays:

- The status of the Oracle ASM instance
- A chart that shows the used and free space of each disk group
- A list of databases and Oracle ASM Cluster File Systems serviced by the Oracle ASM instance

Automatic Storage Management: +ASM_dbhost.example.com

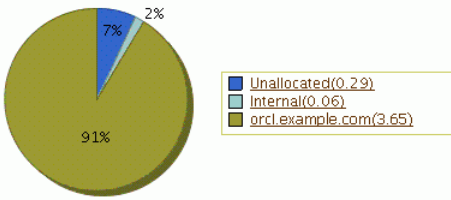
Home Performance Disk Groups Configuration Users ASM Cluster File System

Data Retrieved May 1, 2009 12:11:53 PM PDT Refresh

General

Current Status **Up**
 Up Since **Apr 24, 2009 11:59:16 PM PDT**
 Availability **100**
 (Last 24 hours)
 Instance Name **+ASM**
 Version **11.2.0.0.2**
 Host [dbhost.example.com](#)
 Oracle Home [/u01/app/oracle/product/11.2.0/grid](#)

Disk Group Usage (GB)



Unallocated(0.29)
 Internal(0.06)
 orcl.example.com(3.65)

Diagnostic Summary

Alert Log [No ORA- errors](#)
 Active Incidents **0**

Serviced Databases

Name	Disk Groups	Failure Groups	Allocated Space (GB)	Availability	Alerts
orcl.example.com	DATA	n/a	3.65		1 1

Serviced ASM Cluster File Systems

Mount Point	Availability	State	Used (%)	Used (GB)	Size (GB)	Allocated Space (GB)	Volume	Disk Group
(No ASM Cluster File Systems)								

Other Volumes

Volume	Volume Device	Usage	State	Disk Group	Size (GB)	Allocated Space (GB)	Redundancy
No object found							

Alerts

Severity	Category	Name	Impact	Message	Alert Triggered
	Disk Group Usage	Disk Group Used % of Safely Usable		Disk group DATA has used 92.87% of safely usable free space.	May 1, 2009 12:08:01 PM

The procedure for accessing the Oracle ASM Home page differs depending on whether your database is a single-instance database or an Oracle Real Application Clusters database. Details are provided in the following sections:

- ["Accessing the Oracle ASM Home Page on a Single-Instance System"](#) on page A-6

- ["Accessing the Oracle ASM Home Page on an Oracle RAC System"](#) on page A-6

See Also:

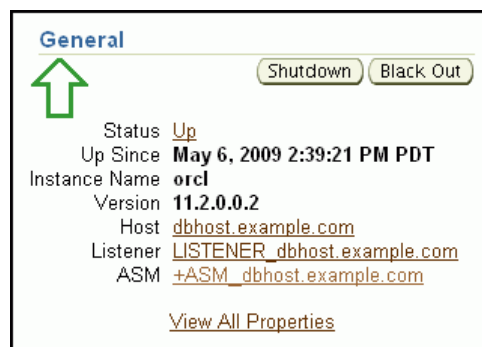
- ["About Oracle Automatic Storage Management"](#) on page A-1

Accessing the Oracle ASM Home Page on a Single-Instance System

On a single-instance system, you access the Oracle ASM Home page starting from the Database Home page.

To access the Oracle ASM Home page on a single-instance system:

1. Go to the Database Home page, logging in as any database user.
See ["Accessing the Database Home Page"](#) on page 3-6.
2. Under the General heading, click the link next to the label ASM.



3. If prompted for Oracle ASM login credentials, then enter the user `SYS`, provide the `SYS` password that was set for the Oracle ASM instance upon installation, and connect as `SYSASM`.

Accessing the Oracle ASM Home Page on an Oracle RAC System

On an Oracle RAC system, you access the Oracle ASM Home page starting from the Cluster Database Home page.

To access the Oracle ASM Home page on an Oracle RAC system:

1. Log in to Oracle Enterprise Manager on any node that is running the Database Control OC4J instance.

The Database Control OC4J instance is automatically started on the node where DBCA was run to create the cluster database. Depending on your configuration, the Database Control OC4J instance may also be running on other nodes.

See ["Accessing the Database Home Page"](#) on page 3-6.

2. On the Cluster Database Home page, under the Instances heading, click the link for the desired Oracle ASM instance.

Monitoring Disk Group Space Usage

Using Oracle Enterprise Manager, you can monitor the total capacity of your disk groups, including the amount of unused space and the amount of unused space that can be safely utilized after taking mirroring into account.

To view Oracle Automatic Storage Management (Oracle ASM) disk group usage and free space:

1. Go to the Oracle ASM Home page.
See "[Accessing the Oracle Automatic Storage Management Home Page](#)" on page A-4.
2. Click **Disk Groups** to view the Disk Groups subpage.
3. If the Oracle ASM Login page appears, then log in as the SYS user, connecting as SYSASM. Provide the SYS password that was set when the Oracle ASM instance was created.

The Disk Groups subpage displays all disk groups with their space usage information.

Automatic Storage Management: +ASM_dbhost.example.com

Home Performance **Disk Groups** Configuration Users ASM Cluster File System

Mount Dismount Rebalance Check Delete

Select All | Select None

Select	Name	State	Redundancy	Size (GB)	Used (GB)	Used (%)	Usable Free (GB)	Member Disks
<input type="checkbox"/>	DATA	MOUNTED	EXTERN	4.00	2.70	67.58	1.30	4
<input type="checkbox"/>	RECOVERY	MOUNTED	EXTERN	4.00	1.72	43.12	2.28	4

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.
 TIP Mount All and Dismount All operation will only mount and dismount the disk groups specified in the Auto Mount Disk Groups parameter.

Home Performance **Disk Groups** Configuration Users ASM Cluster File System

Note: The Usable Free column displays the space in gigabytes (GB) 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 Also:

- "[Overview of Disks, Disk Groups, and Failure Groups](#)" on page A-3

Creating a Disk Group

You may want to create additional Oracle Automatic Storage Management (Oracle ASM) disk groups to do any of the following:

- 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 Serial Advanced Technology Attachment (SATA) drives) into different disk groups. Disks in a disk group should have similar size and performance characteristics.
- Store the fast recovery area in a separate disk group from the database.

To create a disk group:

1. Go to the Oracle ASM Home page.

See "[Accessing the Oracle Automatic Storage Management Home Page](#)" on page A-4.

2. Click the **Disk Groups** link to display the Disk Groups subpage.
3. If the Oracle ASM Login page appears, then log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set when the Oracle ASM instance was created.
4. Click **Create**.

The Create Disk Group page appears. It displays a list of Oracle ASM disks that are available to be added to a disk group. You can use the Select Member Disks drop-down list to choose whether All Disks or Only Candidate disks are displayed.

The screenshot shows the 'Create Disk Group' page. At the top, there is a 'Name' input field and buttons for 'Show SQL', 'Cancel', and 'OK'. Below this, there are radio buttons for 'Redundancy' (HIGH, NORMAL, EXTERNAL) and a dropdown for 'Allocation Unit (MB)'. A note explains that the Allocation Unit (AU) is the fundamental unit of disk space allocation. Below the note, there are four columns labeled 'Member Disks' (1, 2, 3, 4). A dropdown menu 'Select Member Disks' is set to 'Only Candidate Disks'. A table lists the available disks:

Select	Path	Header Status	Library	Label	ASM Disk Name	Size	Unit	Failure Group	Force Reuse
<input type="checkbox"/>	/dev/ASMDisks/disk05	CANDIDATE	SYSTEM			256	MB		<input type="checkbox"/>
<input type="checkbox"/>	/dev/ASMDisks/disk06	CANDIDATE	SYSTEM			256	MB		<input type="checkbox"/>
<input type="checkbox"/>	/dev/ASMDisks/disk07	FORMER	SYSTEM			1024	MB		<input type="checkbox"/>

5. Enter the disk group name, and select redundancy type (high, normal, or external).
6. Select disks as follows:
 - a. Select the check box to the left of each disk to include in the new disk group.
 - b. To force the inclusion of a disk in the disk group, select the **Force Reuse** check box for that disk.

Caution: The Force Reuse check box causes the disk to be added to the new disk group even if the disk 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.

- c. Optionally enter an Oracle ASM disk name for each selected disk. (Oracle ASM provides a name if you do not.)
7. Click **OK** to create the disk group.

See Also:

- *Oracle Automatic Storage Management Administrator's Guide* for a discussion of the FORCE option
- *Oracle Database Reference* for information about the various header status types of Oracle ASM disks
- "[Overview of Disks, Disk Groups, and Failure Groups](#)" on page A-3

Dropping a Disk Group

When you drop a disk group, you delete the disk group, and all of its files. You cannot drop a disk group if any one of its database files is 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. Go to the Oracle ASM Home page.
See "[Accessing the Oracle Automatic Storage Management Home Page](#)" on page A-4.
2. Click the **Disk Groups** link to display the Disk Groups subpage.
3. If the Oracle ASM Login page appears, then log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set when the Oracle ASM instance was created.
4. In the **Select** column, select the name of the disk group to drop.
5. Click **Delete**.
A confirmation page appears.
6. To delete the disk group even if it still contains files, expand **Show Advanced Options** and ensure that the **Including Contents** option is selected.
If the disk group contains files and the **Including Contents** option is not selected, then you cannot drop the disk group.
7. On the confirmation page, click **Yes**.

See Also:

- "[Overview of Disks, Disk Groups, and Failure Groups](#)" on page A-3

Adding Disks to a Disk Group

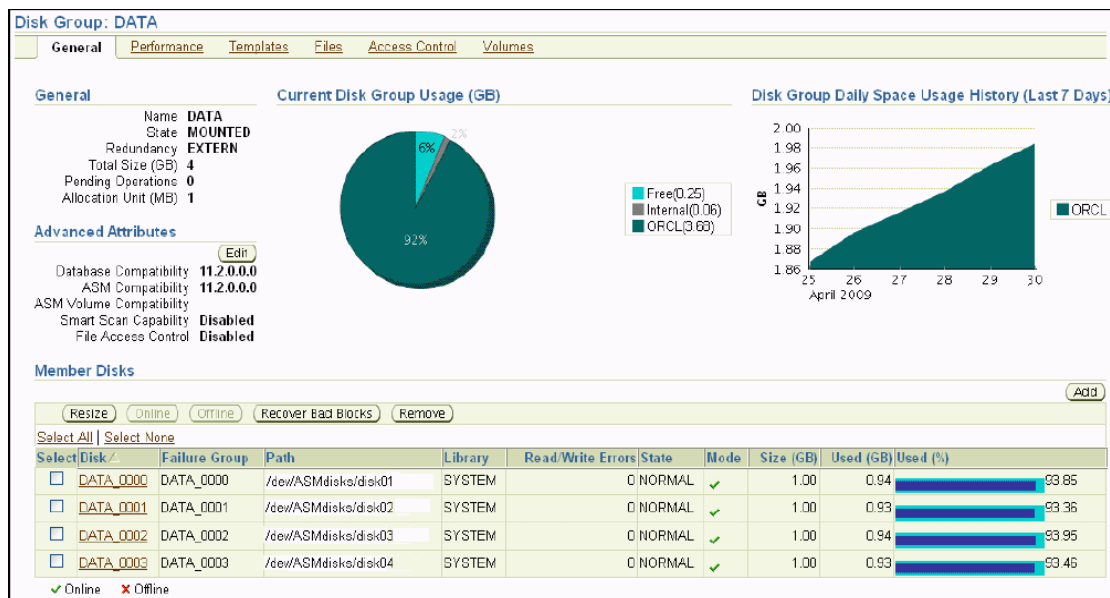
You add disks to a disk group to increase the total amount of storage space in a disk group. You can add one or multiple disks in a single operation. Oracle ASM then rebalances the disk group so that data is evenly distributed on all disks, including the newly added disks.

You can specify a number to control the **power** of the rebalance operation. (The range of valid values depends upon the Oracle Database release you are using. See *Oracle Automatic Storage Management Administrator's Guide*.) The higher the number, the faster the rebalance operation completes. Lower numbers cause rebalancing to take longer, but use 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 prevents the rebalance operation from happening. Manual or automatic rebalancing can then occur at a later time. For example, 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.

To add one or more disks to a disk group:

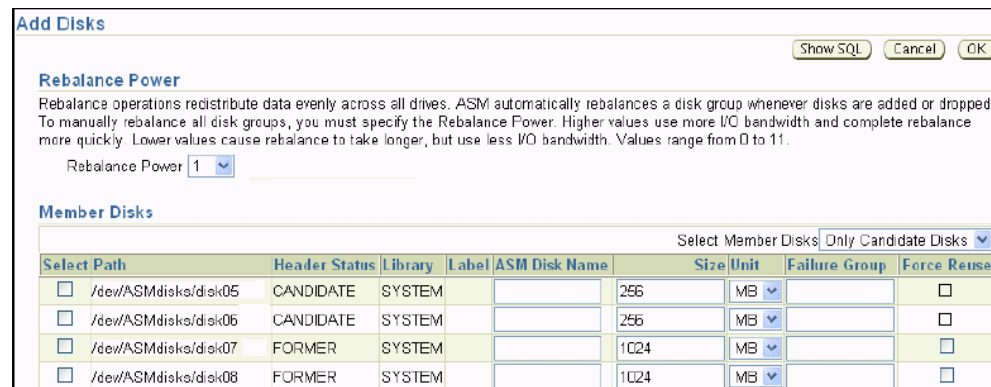
1. Go to the Oracle ASM Home page.
 - See "[Accessing the Oracle Automatic Storage Management Home Page](#)" on page A-4.
2. Click the **Disk Groups** link to display the Disk Groups subpage.
3. If the Oracle ASM Login page appears, then log in as the SYS user, connecting as SYSASM. Provide the SYS password that was set when the Oracle ASM instance was created.
4. 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 in the disk group.



5. Click **Add**.

The Add Disks page appears. It displays a list of Oracle 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. To display not only disks that are available to be added to a disk group, but all Oracle ASM disks, including disks that belong to a disk group (header status = MEMBER), select **All Disks** from the Select Member Disks list.
The page is refreshed and now displays the complete list of Oracle ASM disks that were discovered by the Oracle ASM instance.
7. Optionally change the rebalance power by selecting from the Rebalance Power list.
8. Select disks as follows:
 - a. Select the check box to the left of each disk to add to the disk group.
 - b. 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 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.

- c. Optionally enter an Oracle ASM disk name for each disk. (Oracle ASM provides a name if you do not.)
9. Click **OK** to add the selected disks.

See Also:

- ["Overview of Disks, Disk Groups, and Failure Groups"](#) on page A-3
- *Oracle Automatic Storage Management Administrator's Guide* for more information about controlling rebalance operations and about the FORCE option

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. Oracle 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 must 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. Go to the Oracle ASM Home page.

See "[Accessing the Oracle Automatic Storage Management Home Page](#)" on page A-4.

2. Click the **Disk Groups** link to display the Disk Groups subpage.
3. If the Oracle ASM Login page appears, then log in as the `SYS` user, connecting as `SYSASM`. Provide the `SYS` password that was set when the Oracle ASM instance was created.
4. 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 to drop, and then click **Remove**.

A confirmation page appears.

6. To change rebalance power, or to set the `FORCE` option for dropping disks, do the following:
 - a. Expand **Show Advanced Options**.
 - b. Optionally select the **Force** option or select a value for rebalance power in the Rebalance Power list.

Note: You may need the `FORCE` option if Oracle ASM cannot read from or write to the disk.

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 Also:

- "[Overview of Disks, Disk Groups, and Failure Groups](#)" on page A-3
- *Oracle Automatic Storage Management Administrator's Guide* for more information about dropping disks

Backing Up Oracle ASM-Managed Files

Oracle recommends that you use Recovery Manager (RMAN) to back up files managed by Oracle ASM. See [Chapter 9, "Performing Backup and Recovery"](#) for the information about using RMAN for performing backups.

See Also:

- "[About Oracle Automatic Storage Management](#)" on page A-1

Oracle Automatic Storage Management: Oracle By Example Series

Oracle By Example (OBE) has a series on the *Oracle Database 2 Day DBA* guide. This OBE steps you through the tasks in this appendix and includes annotated screenshots.

To view the Oracle Automatic Storage Management OBE, in your browser, enter the following URL:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/2day_dba/asm/asm.htm

Index

A

adding

- columns to a table, 8-15
- disks to Oracle ASM disk groups, A-10
- table constraints, 8-16

ADDM

See Automatic Database Diagnostics Monitor

administering

- Oracle ASM, A-3
- users, 7-11

administrative user accounts, 7-4

- SYS, 5-7, 7-5
- SYSTEM, 5-7

advisors

- Automatic Database Diagnostics Monitor (ADDM), 10-24
- Buffer Cache Advisor, 10-24
- description, 10-23
- Memory Advisor, 10-24
 - using, 10-38
- performance, 10-23
- PGA Advisor, 10-24
- repairing critical errors, 11-13
- Segment Advisor, 10-24
 - using, 6-23
- SGA Advisor, 10-24
- Shared Pool Advisor, 10-24
- SQL Access Advisor, 10-24
 - using, 10-25
- SQL Tuning Advisor, 10-24
 - using, 10-33
- Undo Advisor, 6-34, 10-24

alerts

- reviewing critical errors, 11-4
- tablespace space usage, 6-19

ALTER SYSTEM statement, 5-15

archived redo log files

- advantages of using, 2-14
- description of, 2-14, 6-5
- viewing, 6-12

ARCHIVELOG mode, 2-13

archiving, 2-14

ARC*n* processes, 5-4

ASM. *See* Oracle ASM

autoextend

data files and tablespaces, 6-9

Automatic Database Diagnostics Monitor (ADDM)

- description, 10-24
- findings, 10-2
- modifying snapshot frequency and retention, 10-22
- overview, 10-2
- snapshots, 10-2
- viewing performance analysis, 10-15

Automatic Diagnostic Repository, 11-1

automatic memory management, 2-15, 5-16

- enabling, 5-18
- modifying settings, 5-19

automatic PGA memory management

- modifying settings, 5-25

automatic shared memory management, 2-15, 5-16

- modifying settings, 5-22

Automatic SQL Tuning Advisor

- about, 10-26
- configuring, 10-27
- viewing results, 10-30

Automatic Storage Management. *See* Oracle ASM

Automatic Workload Repository, 10-8

AWR

See Automatic Workload Repository

B

background processes

- and database configuration, 2-16
- archiver (ARC*n*), 5-4
- checkpoint, 5-4
- database writer (DBW*n*), 5-3
- description, 1-2, 5-3
- log writer (LGWR), 5-3
- manageability monitor (MMON), 5-4
- process monitor (PMON), 4-2, 5-4
- system monitor (SMON), 5-4
- use by database instances, 5-1

backup

- available, 9-24
- backup pieces, 9-9
- channels, 9-9
- consistent, 9-3
- cross-checking, 9-24, 9-25
- deleting expired backups, 9-26

- deleting obsolete backups, 9-27
- expired, 9-24
- files, 6-11
- full backup of data files, 9-12
- image copies, 9-9
- inconsistent, 9-3
- incremental backup of data files, 9-13
- making unavailable, 9-26
- managing, 9-23
- obsolete, 9-24
- Oracle suggested backup strategy, 9-17
- overview, 9-1
- sets, 9-9
- settings, 9-9
- tags, 9-13
- unavailable, 9-24
- validating, 9-19
- whole database backup, 9-14
- backup and recovery area
 - specifying during database creation, 2-14
- BasicFiles, 8-7
- blackouts
 - defining time periods, 3-11
 - description, 3-11
- block size
 - configuring for the database, 2-16
- Buffer Cache Advisor
 - description, 10-24

C

- central database management
 - configuring, 2-13
- character sets, 2-5
 - defining, 2-16
 - description, 2-16
- checkpoint
 - of the database, 5-4
- checkpoint process, 5-4
- classes, Java, 8-30
- CLOB data type, 8-5
- code
 - database-resident, 8-30
- columns
 - adding, 8-15
 - deleting, 8-15
- comments
 - adding to problem activity logs, 11-12
- compiling schema objects, 8-31
- compressed tables, 6-24, 8-8
- configuring
 - Automatic SQL Tuning Advisor, 10-27
- connect descriptors
 - description, 4-3
 - example, 4-3
- CONNECT role, 7-3
- connect strings
 - description, 4-3
- connection modes
 - configuring, 2-18

- connection privileges
 - SYSASM, A-3
 - SYSDBA, 5-7
 - SYSOPER, 5-7
- constraints, 8-6
 - adding, 8-16
 - deleting, 8-18
 - modifying, 8-18
- control files
 - description, 6-3
 - multiplexing, 6-3
 - SQL*Loader, 8-19
- creating
 - databases
 - overview, 2-1
 - steps, 2-6
 - using standalone DBCA, 2-10
 - indexes, 8-24
 - service requests, 11-8
 - tables, 8-11
 - users, 7-13
- customized repairs, 9-31

D

- data
 - loading, 8-19
- data blocks
 - description of, 2-16, 6-6
- data dictionary tables
 - access to as user SYS, 7-6
- data files
 - and the Oracle Suggested Backup Strategy, 9-17
 - autoextend, 6-9, 6-18
 - autoextend and Oracle Flashback, 6-31
 - backing up, 9-3
 - backups of only used blocks, 9-9
 - data blocks, 6-6
 - description of, 6-6
 - example of corrupt block detection, 9-30
 - extents, 6-6
 - full backups, 9-12
 - image copies of, 9-12
 - media recovery, 9-3
 - missing, 9-2
 - recovering from the loss of, 9-38
 - recovery without restoration, 9-2
 - restoring from consistent and inconsistent backups, 9-3
 - segments, 6-6
 - SQL*Loader, 8-19
 - tempfiles, 6-6
 - validating backups of, 9-21
 - viewing information about, 6-13
- data integrity, 8-6
- Data Recovery Advisor, 9-28
- data types
 - CLOB, 8-5
 - DATE, 8-5
 - NUMBER, 8-5

- overview, 8-4
- user-defined, 8-7
- VARCHAR2, 8-5
- database
 - domain, 2-5
 - global name, 2-5
 - management options, 2-6
- database administration
 - privileges required, 7-4
- database checkpoint, 5-4
- Database Configuration Assistant (DBCA)
 - configuring database options, 2-18
 - deleting a databases, 2-19
 - managing database templates, 2-19
 - using to create a database, 2-10
- database configuration options, 2-5
- Database Control
 - See* Oracle Enterprise Manager Database Control
- database edition, 2-3
- database links
 - description, 8-34
- database objects
 - naming, 8-2
- Database Upgrade Assistant (DBUA)
 - overview, 12-2
 - restrictions on versions, 12-3
 - starting, 12-4
 - steps for upgrading, 12-5
- database-resident program code, 8-30
- databases
 - advisors
 - using, 10-23
 - alerts, 10-1
 - responding to alerts, 10-13
 - setting metric thresholds, 10-11
 - setting notification rules, 10-14
 - viewing metrics, 10-8
 - backup
 - description, 9-3
 - backup and recovery, 9-1
 - configuring options, 2-18
 - creating with DBCA, 2-10
 - deleting using DBCA, 2-19
 - diagnosing performance problems, 10-15
 - duplicating, 2-19
 - managing
 - roadmap, 3-1
 - with Oracle Enterprise Manager Database Control, 3-1
 - managing with Oracle Enterprise Manager Database Control, 3-1
 - metric thresholds, 10-1
 - monitoring
 - alert-based, 10-1
 - performance, 10-1
 - overview of creating, 2-1
 - performance
 - advisors, 10-23
 - monitoring, 10-1
 - recovery
 - description, 9-3
 - point-in-time, 9-4
 - restoring
 - description, 9-3
 - starting and stopping on Microsoft Windows, 5-11
 - steps for installing and creating, 2-6
 - storage structure
 - logical structures, 6-1
 - physical structures, 6-1
 - tuning, 10-1
 - upgrading, 12-2, 12-5
 - users, 7-1
 - using DBCA templates, 2-19
- DATE data type, 8-5
- DBA role, 7-4, 8-2
- DBUA
 - See* Database Upgrade Assistant
- DBWn processes, 5-3
- dedicated server mode, 2-18
- dedicated server processes, 5-4
- deinstall tool
 - location, 12-10
 - syntax and options, 12-10
- deleting
 - constraints, 8-18
 - databases, 2-19
 - table columns, 8-15
 - tables, 8-21
 - users, 7-22
 - views, 8-30
- directory
 - Oracle base, 2-4
 - Oracle inventory, 2-4
- disk discovery
 - in Oracle ASM, A-3
- disk groups
 - adding disks to, A-10
 - creating, A-8
 - dropping, A-10
 - dropping disks from, A-13
 - monitoring disk space usage, A-7
 - overview, A-3
- dropping
 - disk groups, A-10
 - disks from disk groups, A-13
 - tablespaces, 6-22
 - undoing with Flashback Drop, 9-35
- duplicating
 - databases, 2-19
 - users, 7-15

E

- emctl, 3-5
- enabling
 - automatic memory management, 5-18
- Enterprise Edition, 2-3
- Enterprise Manager
 - See* Oracle Enterprise Manager Database Control

- EXAMPLE tablespace, 6-7
 - configuring, 2-15
- expiring passwords
 - reasons for, 7-21
- extending
 - tablespaces, 6-9, 6-18
 - tablespaces and Oracle Flashback, 6-31
 - undo tablespaces, 6-31
- extents
 - description of, 6-6

F

- failure groups, A-3
- fast recovery area, 9-4
 - configuring, 9-7
 - retention policies, 9-6
 - specifying during database creation, 2-14
- Flashback Database, 9-37
- Flashback Drop, 9-34
- flashback features
 - Flashback Database, 9-37
 - Flashback Drop, 9-34
 - Flashback Table, 9-32
- Flashback Table, 9-32, 9-33
- functions, PL/SQL, 8-30

G

- global database name, 2-5
- global indexes, 8-23
- granting
 - privileges, 7-16
 - roles, 7-16
- grid control
 - configuring, 2-13

I

- inactive, 6-29
- incidents
 - about, 11-2
 - closed, viewing, 11-16
 - closing, 11-14
 - tracking, 11-2
- indexes
 - creating, 8-24
 - description, 8-21
 - global, 8-23
 - viewing, 8-24
- initialization parameter file, 6-10
- initialization parameters
 - description, 5-2
 - how they are used by the database, 5-2
 - server parameter files, 5-2
 - viewing and modifying, 5-12
- installation
 - checking prerequisites, 2-2
 - choices, 2-2
 - overview, 2-1
 - software location, 2-4

- steps, 2-6
- storage options, 2-4
- using DBCA for database, 2-10

- instances
 - database instances, 5-1
 - management, 5-2
 - memory structure, 5-5
 - Oracle instances, 5-1
 - PGA
 - definition, 5-6
 - shutdown, 5-8
 - startup, 5-7
- invalid schema objects, 8-30

J

- Java classes, 8-30
- Java source code, 8-30

L

- language
 - used by software, 2-5
- LGWR process, 5-3
- licensed software options, 2-3
- listener.ora files, 4-2
- listeners
 - See* Oracle listeners
- Load Data wizard
 - using, 8-20
- loading data, 8-19
- LOB columns, 8-7
- local database management
 - configuring, 2-13
- locally managed tablespaces, 6-8
- locking and unlocking users, 7-21
- log switch
 - description, 2-13

M

- managing
 - memory, 5-15
- media recovery, 9-3
- memory
 - configuring, 2-15
 - management, 5-5, 5-15
 - requirements, 2-2
 - structure of in an Oracle instance, 5-5
- Memory Advisors
 - description, 10-24
 - using, 10-38
- mirroring, A-2
- MMON process, 5-4
- modifying
 - initialization parameters, 5-12
 - table attributes, 8-14
 - table constraints, 8-18
 - users, 7-20

N

Net Configuration Assistant (NETCA)

description, 4-4

NETCA

See Net Configuration Assistant (NETCA)

network configuration

connection requests, 4-2

description, 4-1

notification of alerts

alert notification, 3-11

NUMBER data type, 8-5

O

object privileges, 7-3

offline tablespaces, 6-9, 6-21

online redo log files, 6-29

current, 6-29

description of, 2-13, 6-4

multiplexing, 6-4, 6-28

switching, 6-29

viewing, 6-12

online redo log groups

description of, 2-13

online table redefinition, 6-22

operating system groups, 2-6

Oracle ASM, A-1

administering, A-3

definition, A-1

disk discovery, A-3

failure groups, A-3

installation, 2-4

mirroring, A-2

monitoring disk group space usage, A-7

rebalancing, A-2

Oracle Automatic Storage Management. *See* Oracle ASM.

Oracle base directory, 2-4

Oracle Enterprise Manager Database Control

backup, 9-14

backup and recovery, 9-5

configuration information, 2-10

description, 3-2

features, 3-3

installing, 2-6

managing databases, 3-1

online help, 3-3

starting and stopping, 3-5

Oracle inventory directory, 2-4

Oracle listeners

listener.ora file, 4-2

starting

Oracle Net

description, 4-1

listener configuration, 4-2, 4-5

mapping methods, 4-4

directory naming, 4-4

easy connect naming, 4-4

local naming, 4-4

Oracle suggested repair, 9-28

Oracle system identifier, 2-5

ORACLE_BASE environment variable, 2-8, 2-9

ORACLE_HOME environment variable, 2-8

ORACLE_HOSTNAME environment variable, 2-2

P

parameter file, 6-10

partitioned tables, 8-7

password file, 6-10

password policies

default, 7-23

description of, 7-23

setting, 7-23

passwords

expiring, 7-21

patch sets, 12-1

patches, 12-1

PGA

See Program Global Area

PGA Advisor

description, 10-24

PMON process, 4-2, 5-4

predefined roles, 7-3

privileges

administrative, 7-4

and synonyms, 8-34

authenticating SYSDBA and SYSOPER

users, 6-10

connection, 5-7

for installation of Oracle Database software, 2-7

granting, 7-16

granting to a Database Control administrative

user, 3-9

object, 7-3

recommended, 7-1

required for database administration, 3-8, 7-4

revoking a system, 7-17

schema object management, 8-2

SYSASM, A-3

SYSDBA, 5-7, 9-5

SYSOPER, 5-7

system, 7-3

users, 7-3

using roles to manage, 7-3

problem activity logs

adding comments to, 11-12

problems

about, 11-1

adding comments to activity logs, 11-12

problem key, 11-1

viewing, 11-14

viewing detail, 11-7

procedures, 8-30

profiles, 7-23

Program Global Area (PGA)

components, 5-6

definition, 5-6

description, 5-6

target setting, 10-38

Q

quotas

assigning for a tablespace, 7-19

R

read consistency, 6-30

read only tablespaces, 6-9

read write tablespaces, 6-9

rebalancing, A-2

reclaiming space, 6-22

recovery

complete, 9-4

configuration, 2-13

fast recovery area, 9-4

incomplete, 9-4

media, 9-3

overview, 9-1

whole database recovery, 9-37, 9-38

Recovery Manager (RMAN)

channels, 9-9

control file use, 9-4

overview, 9-2

parallelism, 9-10

recovery catalog, 9-4

repository, 9-4

recycle bins, 8-21

redo log files

See online redo log files *or* archived redo log files

reference collection, 10-6

reorganizing tables, 6-22

repairs

customized, 9-31

Oracle suggested, 9-28

RESOURCE role, 7-4, 8-2

revalidating schema objects, 8-31

revoking

roles, 7-17

system privileges, 7-17

roles

administering, 7-6

CONNECT, 7-3

DBA, 7-4, 8-2

granting, 7-16

predefined, 7-3

RESOURCE, 7-4, 8-2

revoking, 7-17

using to manage user privileges, 7-3

viewing, 7-6

rollback, 6-30

rollback segments

description, 6-5

S

sample schemas

configuring, 2-15

installation of, 2-5

schema objects

compiling, 8-31

database links, 8-34

description of, 8-1

indexes, 8-21

creating, 8-24

viewing, 8-24

invalid, 8-30

privileges, 8-2

sequences, 8-33

synonyms, 8-33

tables, 8-3

constraints, 8-6

modifying, 8-14

validating, 8-31

views, 8-27

schemas, 7-2

SecureFiles, 8-7

security options, 2-5

Segment Advisor

description, 10-24

description of, 6-23

manual operation, 6-28

recommendations, 6-24

viewing recommendations, 6-24

segments

description of, 6-6

sequences

description, 8-33

server parameter files, 5-2

server processes

description, 5-4

service requests

creating, 11-8

tracking, 11-12

services

on Microsoft Windows, 5-11

session

description of, 5-6

setting

password policies, 7-23

SGA

See System Global Area

Shared Pool Advisor

description, 10-24

shared server mode, 2-18

shared server processes, 5-4

shrinking segments, 6-22

shutting down databases

Microsoft Windows, 5-11

SID, 2-5

SMON process, 5-4

snapshots, 10-2

source code, Java, 8-30

space usage alerts for tablespaces, 6-19

SQL Access Advisor

using, 10-25

SQL Advisors

about, 10-25

description, 10-24

SQL Profile, 10-26

SQL Response Time, 10-6

- SQL statements
 - about, 3-15
- SQL Tuning Advisor
 - about, 10-25
 - using, 10-33
- SQL Tuning Set, 10-6
- SQL*Loader
 - control file, 8-19
- Standard Edition, 2-3
- starting databases
 - Microsoft Windows, 5-11
- startup and shutdown
 - about, 5-7
 - privileges required for, 5-7
- subprograms, 8-30
- Support Workbench
 - about, 11-1
 - Custom Packaging process, 11-10
 - home page, 11-3
 - Quick Packaging process, 11-10
 - viewing problems, 11-14
- synonyms
 - and privileges, 8-34
 - description, 8-33
- SYS user, 3-8, 5-7, 7-4
- SYSASM system privilege, A-3
- SYSAUX tablespace, 6-7
- SYSDBA, 2-6
- SYSDBA system privilege, 5-7, 6-10, 7-5
- SYSMAN user, 3-8
- SYSOPER, 2-6
- SYSOPER system privilege, 6-10, 7-5
- System Global Area (SGA)
 - components, 5-5
 - description, 5-5
 - target setting, 10-38
- system identifier (SID), 2-5
- system privileges, 7-3
 - SYSASM, A-3
 - SYSDBA and SYSOPER, 7-5
- SYSTEM tablespace, 6-7
- SYSTEM user, 3-8, 5-7, 7-4

T

- tables
 - adding columns to, 8-15
 - adding constraints, 8-16
 - compressed, 6-24, 8-8
 - creating, 8-11
 - deleting, 8-21
 - deleting columns, 8-15
 - deleting constraints, 8-18
 - description, 8-3
 - loading data into, 8-19
 - LOB columns, 8-7
 - modifying attributes, 8-14
 - modifying constraints, 8-18
 - partitions, 8-7
 - retrieving a dropped table, 9-36

- viewing data in, 8-10
- viewing information about, 8-8
- tablespaces
 - autoextend, 6-9, 6-18
 - autoextend and Oracle Flashback, 6-31
 - creating, 6-14
 - description, 6-7
 - dropping, 6-22
 - EXAMPLE, 6-7
 - extending undo tablespaces, 6-31
 - locally managed, 6-8
 - modifying, 6-18
 - offline, 6-9
 - quotas, 7-19
 - read only and read write, 6-9
 - Segment Advisor, 6-18
 - space usage alerts, 6-19
 - SYSAUX, 6-7
 - SYSTEM, 6-7
 - taking offline, 6-21
 - TEMP, 6-7
 - temporary tablespaces, 6-9
 - types of, 6-8
 - undo tablespaces, 6-8, 6-31
 - UNDOTBS1, 6-7
 - USERS, 6-8
 - viewing information about, 6-14
- TEMP tablespace, 6-7
- templates
 - creating databases with, 2-12
 - using with DBCA, 2-19
- temporary tablespaces, 6-9
- triggers, 8-30

U

- undo
 - about, 6-30
 - advisor, 6-34
 - managing, 6-30, 6-31
 - tablespaces, 6-8
 - viewing, 6-32
- Undo Advisor
 - computing minimum undo tablespace size
 - with, 6-34
 - description, 10-24
 - setting minimum undo retention period
 - with, 6-34
- undo retention, 6-31
- undo retention period
 - setting with Undo Advisor, 6-34
- undo tablespaces
 - autoextend, 6-31
 - changing to fixed-size, 6-37
 - computing minimum size for fixed-size
 - tablespace, 6-34
 - default configuration, 6-31
 - description of, 6-31
- UNDOTBS1 tablespace, 6-7
- upgrading a database, 12-5

- user accounts
 - description, 7-1
- user-defined data types, 8-7
- users
 - accounts
 - administrative, 7-4
 - description, 7-1
 - administering, 7-11
 - creating, 7-13
 - creating by duplication, 7-15
 - deleting, 7-22
 - duplicating, 7-15
 - expiring passwords for, 7-21
 - locking and unlocking, 7-21
 - modifying, 7-20
 - privileges, 7-3
 - SYSASM, A-3
 - viewing, 7-12
- USERS tablespace, 6-8

V

- validating schema objects, 8-31
- VARCHAR2 data type, 8-5
- viewing
 - indexes, 8-24
 - initialization parameters, 5-12
 - table data, 8-10
 - table information, 8-8
 - users, 7-12
 - views, 8-28
- views
 - deleting, 8-30
 - description, 8-27
 - displaying, 8-28

W

- wizards
 - Create Blackout, 3-11
 - database creation, 2-12
 - incident reporting, 11-4, 11-10
 - Load Data wizard, 8-19
 - Oracle ASM migration, A-2
 - Quick Packaging, 11-10
 - Recovery, 9-4, 9-31, 9-37
 - Schedule Customized Backup, 9-19
 - SQL Access Advisor, 10-36