
Enterprise DBA Part 1B: Backup and Recovery Workshop

Volume 2 • Instructor Guide

30050GC10

Production 1.0

August 1999

M09098

ORACLE®

Authors

Dominique Jeunot
Shankar Raman

Technical Contributors and Reviewers

David Austin
Ben van Balen
Ralf Durben
Bruce AErnst
Joel Goodman
Bert van Gorkom
Scott Gossett
Lex de Haan
John Hough Jr.
Alexander Hunold
Peter Kilpatrik
Stefan Lindblad
Hanne Rue Rasmussen
Robert Thome
Jean-Francois Verrier
Steven Wertheimer

Publisher

John B Dawson

Copyright © Oracle Corporation, 1999. All rights reserved.

This documentation contains proprietary information of Oracle Corporation. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c) (1) (ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of the Worldwide Education Services group of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Right," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Education Products, Oracle Corporation, 500 Oracle Parkway, Box 659806, Redwood Shores, CA 94065. Oracle Corporation does not warrant that this document is error-free.

SQL*Loader, SQL*Net, SQL*Plus, Net8, Oracle Call Interface, Oracle7, Oracle8, Developer/2000, Developer/2000 Forms, Designer/2000, Oracle Enterprise Manager, Oracle Parallel Server, Oracle Server Manager, PL/SQL, Pro*C, Pro*C/C++, and Trusted Oracle are trademarks or registered trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

Contents

Preface

Profile	xvii
Related Publications	xix
Typographic Conventions	xx

Introduction

Objectives	I-2
------------	-----

Lesson 1: Backup and Recovery Considerations

Objectives	1-2
Overview	1-4
Defining a Backup and Recovery Strategy	1-5
Business Requirements	1-6
Operational Requirements	1-7
Technical Requirements	1-9
Disaster Recovery Issues	1-11
Oracle Availability and Features	1-13
Summary	1-14

Lesson 2: Oracle Recovery Structures and Processes

Objectives	2-2
Basic Oracle Architecture	2-4
The Large Pool	2-8
Data Buffer Cache, DBW _n , and Data Files	2-11
Redo Log Buffer, LGWR, and Redo Log Files	2-14
Multiplexed Redo Log Files	2-17
CKPT Process	2-19
Fast-Start Checkpointing	2-21
Multiplexing Control Files	2-26
ARC _n Process and Archived Log Files	2-28
Categories of Failures	2-31
Common Causes of Statement Failure	2-32

Resolutions for Statement Failures	2-33
Causes of User Process Failures	2-34
Resolving User Process Failures	2-35
Possible User Error Failures	2-36
Resolving User Errors	2-37
Instance Failure	2-38
Recovery from Instance Failure	2-39
Instance Recovery Process	2-41
Media Failure	2-43
Media Failure Resolution	2-44
Database Synchronization	2-45
Fast-Start Parallel Rollback	2-46
On-Demand Parallel Rollback	2-51
Quick Reference	2-52
Summary	2-53

Lesson 3: Oracle Backup and Recovery Configuration

Objectives	3-2
Redo Log	3-3
NOARCHIVELOG Mode	3-4
ARCHIVELOG Mode	3-6
Set Archivelog Destination	3-8
Duplexing Archived Log Files	3-9
Specifying Multiple Archive Locations	3-10
Multiple Archive Options	3-11
Specifying Minimum Number of Local Destinations	3-13
Controlling Archiving to a Destination	3-15
Enabling ARCHIVELOG Mode	3-16
Multiple Archive Processes	3-18
Enabling Archive Process	3-19
Enabling the Archive Process in an Open Instance	3-21
Enabling Archive Processes at Start of Instance	3-23
Stop or Start Additional Archive Processes	3-24

Disabling Archive Processing	3-25
Selectively Archiving Log Files	3-26
Obtaining Archive Log Information	3-28
Recovery Configuration	3-31
Monitoring Recovery Time	3-32
Summary	3-34
Quick Reference	3-35

Lesson 4: Physical Backups Without Oracle Recovery Manager

Objectives	4-2
Overview	4-4
Closed Database Backups	4-5
Obtaining Database File Information	4-8
Performing a Closed Database Backup	4-10
Opened Database Backup	4-12
Performing an Opened Database Backup	4-16
Data Dictionary Views	4-18
Backing Up a Control File	4-20
Read-Only Tablespace Backup	4-22
Read-Only Tablespaces	4-23
Logging and Nologging Options	4-24
Summary	4-25
Quick Reference	4-26

Lesson 5: Complete Recovery Without Recovery Manager

Objectives	5-2
Overview	5-3
Media Failure	5-4
Recovery: NOARCHIVELOG Mode	5-6
Restoring Data Files	5-7
Complete Recovery	5-8
Recovery by Using Archived Log Files	5-13
Locating Data Files	5-15
Recovery After Failure of Opened Database Backup	5-27

Clearing Corrupted Redo Logs	5-30
Loss of Inactive Redo Log File	5-32
Re-creating Redo Logs	5-34
Recovery Status Information	5-36
Summary	5-38
Quick Reference	5-39

Lesson 6: Incomplete Oracle Recovery with Archiving

Objectives	6-2
Overview	6-3
Reasons for Incomplete Recovery	6-4
Incomplete Recovery	6-5
RECOVER Command	6-7
Recovery Steps	6-8
Incomplete Recovery Guidelines	6-9
The Alert Log	6-11
Time-Based Recovery	6-12
Incomplete Recovery Using Until Time	6-14
Incomplete Recovery Using Until Cancel	6-16
Incomplete Recovery Using the Backup Control File	6-19
Loss of Current Online Redo Logs	6-22
Recovery Through Resetlogs	6-26
Summary	6-29
Quick Reference	6-30

Lesson 7: Oracle Export and Import Utilities

Objectives	7-2
Export and Import Utility Overview	7-3
Methods to Run the Export Utility	7-5
Export Modes	7-6
Command Line Export	7-7
Complete Export	7-9
Incremental Export	7-10
Cumulative Export	7-11

Incremental and Cumulative Export Benefits	7-12
Direct Path Export Concepts	7-13
Specifying Direct Path Export	7-14
Direct Path Export Features	7-15
Direct Path Export Restrictions	7-16
Export Utility Compatibility Issues	7-17
Using the Import Utility for Recovery	7-18
Import Modes	7-19
Command Line Import	7-20
Import Process Sequence	7-22
NLS Considerations	7-24
Tablespace Point-in-Time Recovery (TSPITR)	7-26
TSPITR Methods	7-28
TSPITR by Using Transportable Tablespaces	7-29
Summary	7-31
Quick Reference	7-32

Lesson 8: Additional Recovery Issues

Objectives	8-2
Fast-Start Recovery	8-3
Minimize Downtime	8-5
Parallel Recovery Operations	8-7
Starting a Database with a Missing Data File	8-10
Loss of Non-Essential Data File	8-12
Loss of Control Files	8-15
Recovering Control Files	8-16
Read-Only Tablespace Recovery	8-18
Read-Only Tablespace Recovery Issues	8-19
Summary	8-20
Quick Reference	8-21

Lesson 9: Oracle Utilities for Troubleshooting

Objectives	9-2
The Alert Log File	9-3
Oracle Trace Files	9-5
Corrupt Block Detection and Repair	9-7
Methods for Detecting Block Corruption	9-8

DB_BLOCK_CHECKSUM Parameter	9-10
DB_BLOCK_CHECKING Parameter	9-11
DBVERIFY Utility	9-12
The Command Line Interface	9-14
DBMS_REPAIR Package	9-16
Mark Corrupted Objects	9-19
Skipping Blocks	9-21
Indexes and Corrupt Tables	9-22
Limitations of DBMS_REPAIR	9-24
ANALYZE VALIDATE STRUCTURE Command	9-25
LogMiner Utility	9-27
LogMiner Analysis	9-29
Views	9-36
Limitations	9-37
Summary	9-38
Quick Reference	9-39

Lesson 10: Oracle Recovery Manager Overview

Objectives	10-2
Overview	10-3
Recovery Manager Features	10-4
Recovery Manager Components	10-6
Recovery Manager Packages	10-8
RMAN Setup Considerations	10-9
The Recovery Catalog	10-11
Control File Information	10-13
Connecting Without a Recovery Catalog	10-15
Recovery Manager Modes	10-17
RMAN Commands	10-19
Channel Allocation	10-21
REPORT Command	10-22
REPORT NEED BACKUP Command	10-23
LIST Command	10-24

Stored Scripts	10-26
RUN Command	10-30
Media Management	10-31
Summary	10-33
Quick Reference	10-34

Lesson 11: Oracle Recovery Catalog Creation and Maintenance

Objectives	11-2
Overview	11-3
Creating the Recovery Catalog	11-7
Connecting Using a Recovery Catalog	11-10
Catalog Maintenance	11-12
CHANGE Command	11-15
Deleting a Backup	11-17
Deleting Records of Previous Incarnation	11-19
Recovery Catalog Backup	11-20
Recovering the Recovery Catalog	11-21
Recovery Catalog Reporting	11-22
LIST Command	11-24
Stored Scripts	11-25
RESET DATABASE Command	11-29
Data Dictionary Views	11-31
Summary	11-33
Quick Reference	11-35

Lesson 12: Backups Using Recovery Manager

Objectives	12-2
Overview	12-3
Backup Concepts	12-4
Whole Database Backup	12-6
Recovery Manager Backup Types	12-8
Allocating a Channel	12-9
Tags	12-11
Image Copies	12-12

Image Copy Characteristics	12-13
COPY Command	12-14
Image Copy Process	12-15
Image Copy Parallelization	12-16
Image Copy of All Data Files	12-17
Monitoring the Copy Process	12-18
Operating System Copies	12-20
Backup Sets	12-21
BACKUP Command	12-22
Backup Set Characteristics	12-25
Multiplexed Backup Sets	12-26
Parallelization of Backup Sets	12-27
Backup Piece	12-29
Backup Piece Size	12-30
Data File Backup Process	12-31
Archived Log Backup Sets	12-32
Archived Log Backup	12-33
Full, Incremental, and Cumulative Backups	12-34
Incremental Backups	12-36
Multilevel Incremental Backup	12-38
Incremental Backups	12-39
Cumulative Incremental Backups	12-41
Backup Constraints	12-42
Backup Set Scenarios	12-43
Backups Using Stored Scripts	12-49
Miscellaneous Issues	12-51
Memory Usage by Recovery Manager	12-53
Troubleshooting	12-55
Data Dictionary Views	12-57
Summary	12-58
Quick Reference	12-59

Lesson 13: Restoration and Recovery Using Recovery Manager

- Objectives 13-2
- Restoration and Recovery Using Recovery Manager 13-3
- Restoring a Database in NOARCHIVELOG Mode 13-4
- Restoring Data Files to a Different Location 13-6
- Recovering a Tablespace 13-7
- Relocating a Tablespace 13-9
- Incomplete Recovery of a Database 13-12
- Incomplete Recovery of a Database: Example 13-13
- Restoring a Database to a Previous Incarnation 13-15
- Summary 13-17

Lesson 14: Oracle Standby Database

- Objectives 14-2
- Overview 14-3
- Standby Database Features 14-4
- Standby Database Guidelines 14-6
- Initialization Parameters 14-7
- Creating a Standby Database 14-8
- Managed Recovery Mode 14-10
- Maintaining the Standby Database 14-11
- Read Only Mode of Standby Database 14-12
- Activating the Standby Database 14-14
- Operating a Standby Database 14-16
- Structural Change of Primary Database 14-18
- Refreshing the Control File 14-20
- Nologging Operations at the Primary Database 14-21
- Summary 14-22
- Quick Reference 14-23

Lesson 15: Workshop

- Objectives 15-2
- Workshop Methodology 15-3
- Workshop Approach 15-5
- Business Requirements 15-6
- Resolving a Failure 15-7
- Summary 15-9

Appendix A: Practices

Practice Session Note	A-2
Practice 2-1	A-3
Practice 3-1	A-4
Practice 4-1	A-5
Practice 5-1	A-6
Practice 5-2	A-7
Practice 6-1	A-11
Practice 7-1	A-13
Practice 8-1	A-14
Practice 9-1	A-15
Practice 10-1	A-16
Practice 11-1	A-17
Practice 12-1	A-18
Practice 13-1	A-19

Appendix B: Practice Solutions

Practice Session Note	B-2
Practice 2-1 Solutions	B-3
Practice 3-1 Solutions	B-5
Practice 4-1 Solutions	B-9
Practice 5-1 Solutions	B-12
Practice 5-2 Solutions	B-14
Practice 6-1 Solutions	B-21
Practice 7-1 Solutions	B-27
Practice 8-1 Solutions	B-28
Practice 9-1 Solutions	B-31
Practice 10-1 Solutions	B-34
Practice 11-1 Solutions	B-35
Practice 12-1 Solutions	B-40
Practice 13-1 Solutions	B-42

Appendix C: Hints

Practice 2-1 Hints	C-2
Practice 3-1 Hints	C-4
Practice 4-1 Hints	C-6
Practice 5-1 Hints	C-7

Practice 5-2 Hints	C-8
Practice 6-1 Hints	C-10
Practice 7-1 Hints	C-12
Practice 8-1 Hints	C-13
Practice 9-1 Hints	C-15
Practice 11-1 Hints	C-16
Practice 12-1 Hints	C-18
Practice 13-1 Hints	C-19

Appendix D: Workshop Scenarios

Workshop Scenarios	D-2
Scenario 1: Loss of INACTIVE Online Redo Log Group	D-3
Scenario 2: Loss of CURRENT Online Redo Log Group	D-4
Scenario 3: Loss of Control Files	D-5
Scenario 4: Loss of Media	D-6
Scenario 5: Loss of File Containing Online Rollback Segment	D-7
Scenario 6: Loss of a Data File of System Tablespace	D-8
Scenario 7: Loss of a Non-System, Non-Rollback Segment Data File	D-9
Scenario 8: Recover from User Errors	D-10
Scenario 9: Failure During Hot Backup	D-11
Scenario 10: Configuring a Recovery Catalog	D-12
Scenario 11: Missing Data File with No Backup	D-14
Scenario 12: Loss of a Data File and Missing Archive Log File	D-15
Scenario 13: Loss of Non-Essential Data File When Database Is Down	D-16
Scenario 14: Recover a Lost Data File from Archive Logs	D-17
Scenario 15: Missing Mirrored Online Redo Log Files	D-18
Scenario 16: Loss of a Control File and Read-Only Tablespace	D-19

Appendix E: Worldwide Support Bulletins

Oracle Corporate Support Problem Repository	E-2
---	-----

Oracle Recovery Manager Overview

Objectives

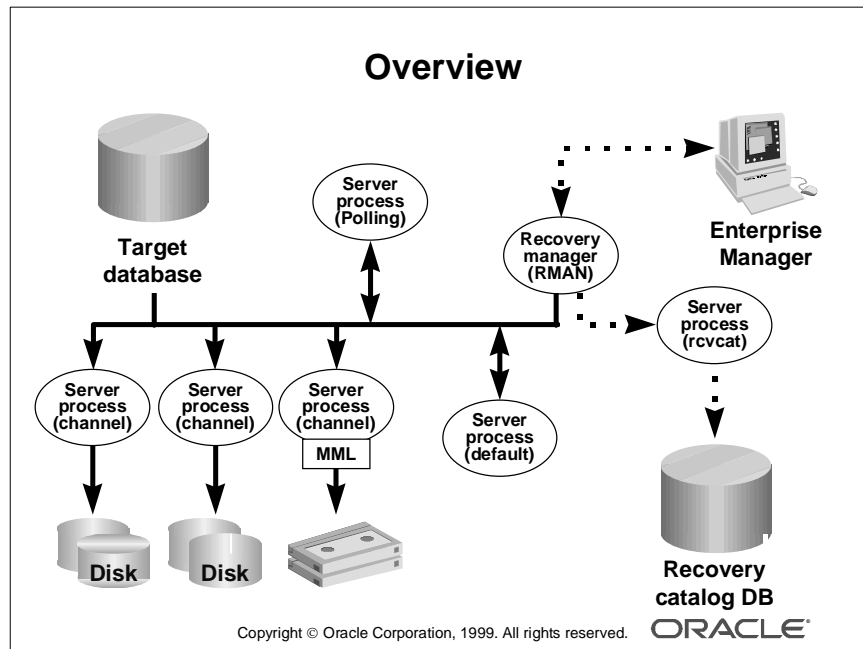
Objectives

After completing this lesson, you should be able to do the following:

- **List the capabilities of Oracle Recovery Manager (RMAN)**
- **Describe the components of RMAN**
- **Connect to Recovery Manager without recovery catalog**
- **Start up and shut down a target database using RMAN**

Copyright ©Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview



Overview

Recovery Manager (RMAN) is a utility available in Oracle RDBMS that helps DBAs manage the backup, restore, and recovery operations on Oracle databases. RMAN has a powerful command language that is independent of the operating system. RMAN uses Oracle server processes for backup and recovery operations, and hence, the backup and recovery operations using RMAN are called Server Managed Backup and Recovery operations.

Recovery Manager has a command line interface. Oracle Enterprise Manager also provides a graphic user interface for the Recovery Manager. Recovery Manager can, however, be used on databases of Oracle8 or higher releases.

Recovery Manager Features

Recovery Manager Features

RMAN provides a flexible way to:

- **Back up the database, tablespaces, datafiles, control files, and archive logs**
- **Store frequently executed backup and recovery operations**
- **Perform incremental block level backup**
- **Compress unused blocks**
- **Specify limits for backups**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE®**

Recovery Manager Features

RMAN provides several features not available when you use operating system managed backup. Some of them are:

- You can store frequently executed operations as scripts in the database.
- Using the incremental block level backups feature you can restrict backup time and size to only those blocks that have changed since the previous backup. This could also help in reducing the time taken to perform recovery operations.
- You can use RMAN to manage the size of backup pieces and save time by parallelizing the backup operation.
- You can verify the backups for their validity.
- You can detect block corruptions. The information relating to the block corruption detected during backup can be obtained by using the dynamic views.
- RMAN operations can be integrated with the scheduling of the operating system to automate backup operations.

Recovery Manager Features

RMAN provides a flexible way to:

- **Detect corrupted blocks during backup**
- **Support Oracle Parallel Server**
- **Increase performance through:**
 - **Automatic parallelization**
 - **Less redo generated**
 - **Restricted I/O for backups**
 - **Tape streaming**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE®**

Recovery Manager Features (continued)

- RMAN provides performance enhancements such as:
 - Automatic parallelization of backup, restore, and recovery operations
 - No generation of extra redo during online database backups
 - Backups restricted to limit reads per file, per second to avoid interfering with OLTP work
 - Prevention of flooding of any one file with reads and writes while still keeping a tape drive streaming, using multiplexing

RMAN has API for use of third-party media management tools to interface with storage devices for speed and reliability.

Recovery Manager Components

Recovery Manager Components

- **Recovery Manager packages**
- **Processes**
- **Target database**
- **RMAN Metadata - Recovery Catalog**
- **Channel**
- **Media management layer**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

Recovery Manager Components

Recovery Manager Executable The Recovery Manager command line interface is invoked through the executable RMAN. RMAN interprets user commands and appropriately invokes server processes to perform the desired tasks.

Processes The server processes invoked by RMAN connect to the target database to perform the backup, restore, and recovery functions through a PL/SQL interface.

Target Database The database for which backup and recovery operations are being performed using the RMAN is called the target database. The control file of the target database contains information about its physical structure, such as the size and location of data files, online and archive logs, and control files. This information is used by the server processes invoked by RMAN in the backup and recovery using RMAN.

Channel To perform and record backup and recovery operations, RMAN requires a link to the target database. This link is referred to as a channel. You must allocate a channel before you begin the execution of backup or recovery commands. The details of how to control and tune the backup and recovery operations through the allocation of channels are discussed in detail in subsequent chapters.

Recovery Manager Components (continued)

Recovery Manager Metadata The data used by RMAN for backup, restore and recovery operations referred to as RMAN metadata is stored in the control file of the target database or when available in a schema of a database. This schema is called the Recovery Catalog. Most of the information stored in the recovery catalog is available in the control file of the target database.

Although it is not mandatory to have a recovery catalog for use of RMAN, it is beneficial to setup a recovery catalog. Many of the features of RMAN, such as stored scripts and automatic backup, are not available without the recovery catalog.

The recovery catalog should be located in a database different from the target database. The creation and maintenance of recovery catalog are discussed in another lesson.

Media Management Layer Media management layer (MML) is used by RMAN when writing to or reading from tapes. The additional media management software required for using the tape medium is provided by media and storage system vendors. The number of additional storage systems supporting RMAN is continuously increasing.

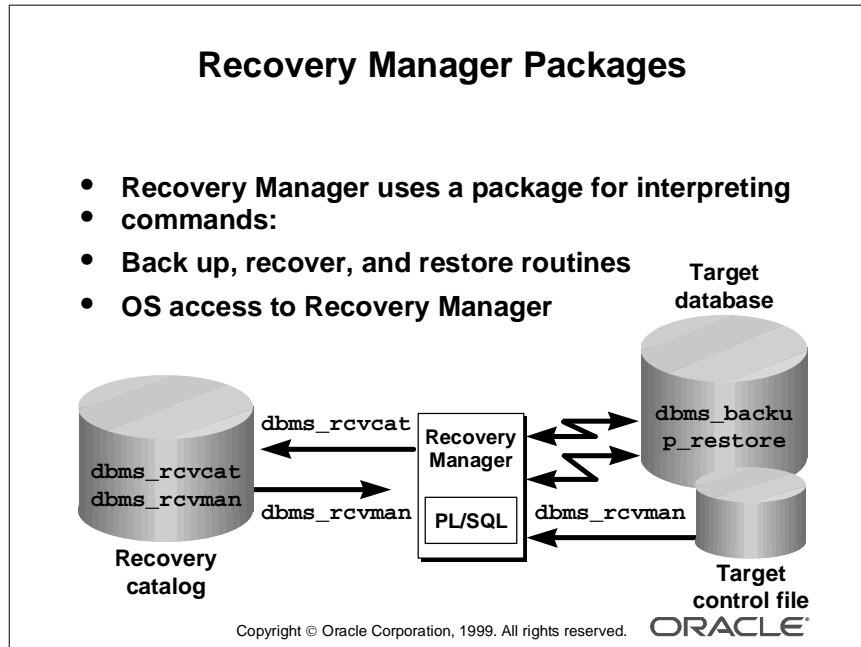
Auxiliary Database Using RMAN, you can create a database that is the duplicate of the target database. Such a duplicate database is called an auxiliary database. The auxiliary database has very high significance when performing tablespace-point-in-time-recovery (TSPITR) using RMAN.

Backup Set When you use RMAN to perform a backup, the output files from one backup operation are referred to as a backup set. Physically, the backup set may comprise of one or more files. Each physical file of a backup set is called a *backup piece*.

Because an Oracle server process is used by RMAN, blocks that are empty are not stored in the backup.

Image Copies Using RMAN, you can copy the database files from one location to another on the disks and these copies can serve the purpose of backup. These copies are called image copies. In an image copy, the empty blocks are also copied over.

Recovery Manager Packages



Recovery Manager Packages

Two sets of packages, `DBMS_RCVCAT` and `DBMS_RCVMAN`, are used by RMAN to perform its tasks. In Oracle8i, these packages are fixed and are a part of Oracle kernel. Hence they are available when the Oracle software is invoked even if the database is in NOMOUNT state. In Oracle 8 these packages were made a part of the data dictionary packages and hence were available only when the target database was in MOUNT state.

`DBMS_RCVCAT` is used by Recovery Manager to maintain information in the recovery catalog, and `DBMS_RCVMAN` queries the control file or recovery catalog.

DBMS_BACKUP_RESTORE Package

This package is created by the `dbmsbkrs.sql` and `prvtbkrs.plb` files. It is used to interface with Oracle and the operating system for creating, restoring, and recovering backups of data files and archive logs.

Once the operating system receives information from the packages, the data is stored on a storage device such as a disk.

RMAN Setup Considerations

RMAN Setup Considerations

- **Resources: Shared memory, more processes**
- **Privileges to users**
 - **Database: SYSDBA SYSOPER**
 - **Operating System: Access to devices**
- **Remote operations**
 - **Set up the password file.**
 - **Ensure that the password file is backed up.**
- **NLS environment variables**
Format used for the time parameters in RMAN commands
- **Use recovery catalog**
 - **More functionality and options**
 - **Maintenance and storage requirements**

Copyright ©Oracle Corporation, 1999. All rights reserved. ORACLE

RMAN Setup Considerations

Before Recovery Manager is used, consider the following points:

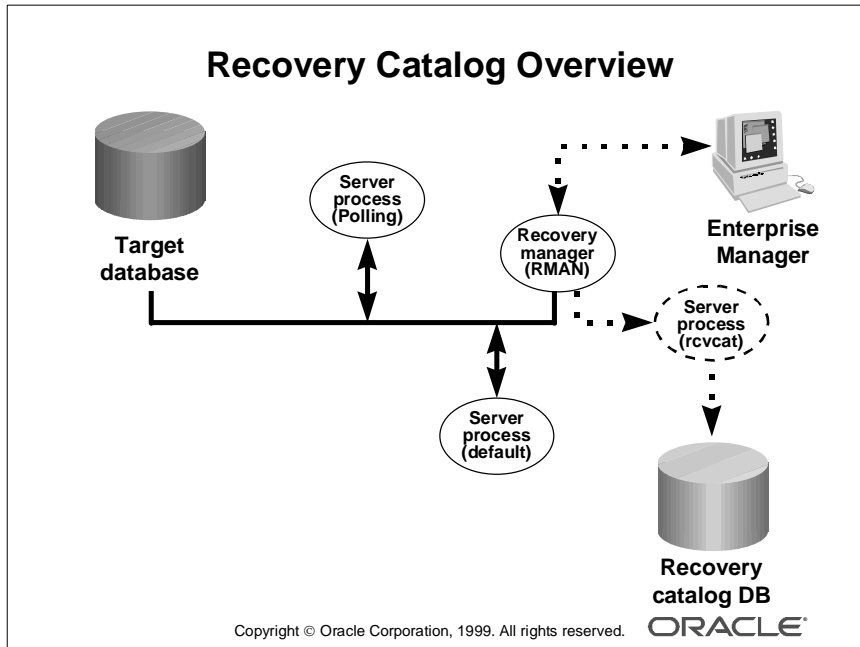
- **Shared resources on the system:** Most of RMAN's work is performed through Oracle server processes. The operations can also be performed in parallel to increase throughput. This implies that the PROCESSES parameter has to be sufficiently high. From the O/S standpoint, this would mean that shared memory and semaphores are adequately set.
- **Set of users performing privileged operations:** You have to decide on the set of users who perform privileged operations. Accordingly, you may set the users' accounts with the necessary privileges at the operating system and at the Oracle database.

For example, if the user performing a backup or restore of a data file does not have the OS privilege to write in the specified directory, the backup or restore operation will fail. Similarly, to start up and shut down a database, the user should have database privilege SYSOPER or SYSDBA.

Recovery Manager Considerations (continued)

- **Remote operations:** You need to use a password file to connect to the target database over Net8 for performing privileged operations such as shutdown, startup, backup, and recovery from a remote machine. You may have to set up the password file. You should ensure that there is a strategy to backup the password file as well.
- **Your NLS environment variables:** Before invoking RMAN, set the NLS_DATE_FORMAT and NLS_LANG environment variables. These variables determine the format used for the time parameters in RMAN commands such as restore, recover, and report.
- **Use of the recovery catalog:** When you use a recovery catalog, RMAN can perform a wider variety of automated backup and recovery functions. Use of the recovery catalog involves storage space and maintenance efforts.
You should also decide if you would have a database dedicated to maintain the recovery catalog of many target databases. Consider also the strategy to back up the recovery catalog.

The Recovery Catalog



Recovery Manager Metadata

RMAN metadata is the information about the target database that RMAN uses to conduct its backup, recovery, and maintenance operations. You can either create a recovery catalog in which to store this information, or let RMAN store it exclusively in the control file of the target database. Although RMAN can conduct all major backup and recovery operations using just the control file, some RMAN commands function only when you use a recovery catalog.

The recovery catalog is maintained solely by RMAN; the target database never accesses it directly. RMAN propagates information about the database structure, archived redo logs, backup sets, and data file copies into the recovery catalog from the control file of the target database.

Recovery Catalog Contents

The recovery catalog is a repository containing information on:

- **Data file and archive log backup sets**
- **Data file copies**
- **Archived redo logs**
- **The physical structure of the target database**
- **Stored job scripts**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Recovery Catalog Contents

The recovery catalog contains information about:

- **Data files and archived log backup sets and backup pieces:**
A backup of either data files or archived logs makes up a backup set. A backup set may have many backup pieces. The catalog stores information such as the name and time of the backup set.
- **Data File copies:**
Copies are physical copies of a data file or archived log stored on a disk. The catalog records the time stamp and name of data file copies.
- **Archived redo logs and copies of them:**
The catalog maintains a record of which archived logs have been created by the database and any copies made by Recovery Manager.
- **The physical structure of the target database:**
It contains similar information as provided by the database control file.
- **Named sequences of commands that call stored scripts:**
Frequently executed commands can be stored in the catalog.

Control File Information

Control File of the Target Database

- The control file holds information used by Recovery Manager.
- The control file can grow depending on the parameter `CONTROLFILE_RECORD_KEEP_TIME`.
- Information in the control file will be recycled.
- The control file cannot store scripts and backup-piece information.

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

Information Obtained from the Control File

The control file contains backup, restore, and recovery actions performed by Recovery Manager. The amount of information grows depending on the frequency of backups, size of the target database, and the retention period.

Configuring the Control File

The `CONTROL_FILE_RECORD_KEEP_TIME` parameter specifies the number of days RMAN information is stored in the control file before being overwritten. A low value results in information being overwritten more frequently, thus minimizing control file growth. If a recovery catalog is used, a lower value should be chosen. The default is 7 days.

Note: When you use a recovery catalog, make sure that resynching is performed more frequently than overwrites to the control file.

For example, if set to 1, Recovery Manager information is kept for a day. If a recovery catalog is used, make sure resynching is performed at least once per day.

Control File Growth

If the size of the control file is too small to store all the information for the time specified by `CONTROL_FILE_RECORD_KEEP_TIME`, then the control file grows. Before the control file grows, certain steps are performed:

- 1 Free space in the control file is used.
- 2 Entries older than `CONTROL_FILE_RECORD_KEEP_TIME` are overwritten.
- 3 If no more space is available, the control file grows as needed.

Control File Copies

Whether or not a recovery catalog is used, Recovery Manager sometimes needs to make a temporary backup of the control file to read. A snapshot control file is created any time RMAN needs to view a read-consistent image of the control file. A read-consistent image is required when querying the noncircular reuse records in the control file. Noncircular reuse records include data file, tablespace, online redo log, and thread information.

The location of this file is configurable:

```
RMAN> set snapshot controlfile name to '/disk1/backup/ctl_300.snp';
```

The default location of the snapshot control file on UNIX systems is:

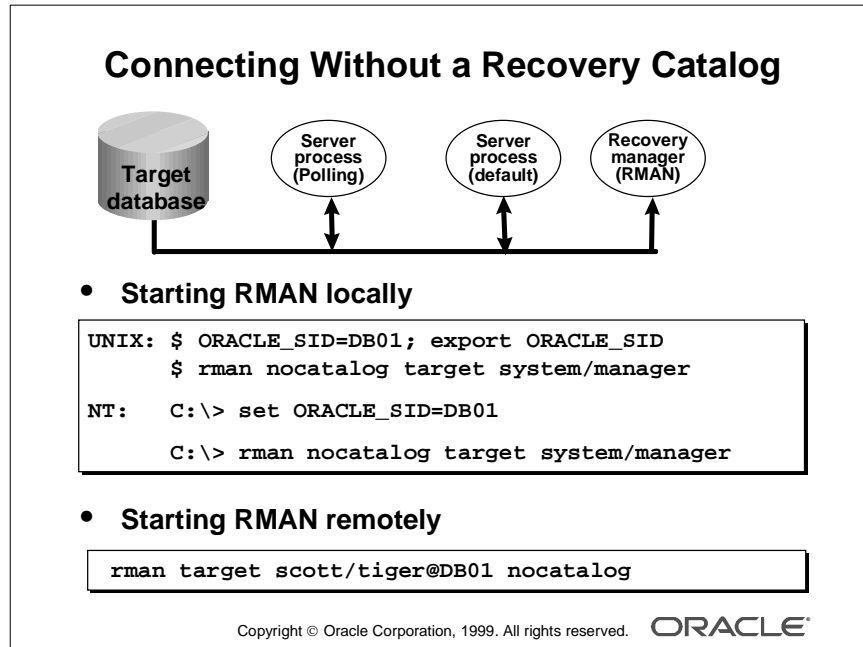
```
$ORACLE_HOME/dbs/snapcf_<dbname>.f
```

where *<dbname>* is the database name.

Re-creating the Control File

The recovery catalog duplicates information contained in the control file. Recovery Manager can use the catalog to re-create lost control files.

Connecting Without a Recovery Catalog



Connecting Without a Catalog

To connect to RMAN without using a recovery catalog, follow these steps:

- 1 Determine which target database you need to back up.
- 2 Use one of the three methods to connect to RMAN.

For a local RMAN connection, at an operating system prompt, type the following:

UNIX

```
$ ORACLE_SID=DB00; export ORACLE_SID
$ rman target system/manager nocatalog
```

NT

```
C:\> SET ORACLE_SID=DB01
C:\> rman target system/manager nocatalog
```

You should connect as a user with SYSDBA privileges on the target database to use RMAN. This helps in performing the privileged operations such as startup and shutdown. To connect from another server, use the TNS alias for the database.

```
rman target system/manager@db01 nocatalog
```

Connecting Without a Catalog (continued)

Considering that it may be easier to access the disk and tape resources locally, you may prefer to login from the server that contains the target database.

Connection Process After you type the RMAN connection command, the following events occur:

- A user process is created for Recovery Manager.
- The user process creates two Oracle Server processes.
 - One default process connected to the target database, for executing SQL commands, resynching the control file, and recovery roll forward
 - One polling process connected to the target database, to locate RPC completions (only one per instance)
 - Backup and recovery information is retrieved from the control file.

Recovery Manager Modes

Recovery Manager Modes

- **A command line interpreter**
- **Interactive mode**
 - Use it when doing analysis.
 - Minimize regular usage.
 - Avoid using with log option.
- **Batch mode**
 - Meant for automated jobs.
 - Minimizes operator errors.
 - Set the log file to obtain information.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

Recovery Manager

Recovery Manager acts as a command line interpreter (CLI) with its own command language. There are two modes of operation with the RMAN—interactive and batch.

Interactive Mode To run RMAN commands interactively, start RMAN and then type commands into the command line interface. For example, you can start RMAN from the UNIX command shell and then execute interactive commands as follows:

```
$ rman target sys/sys_pwd@db1 catalog rman_db1/rman_db1@catdb
RMAN> run {
    2> allocate channel d1 type disk;
    3> backup database;
    4> }
```

Batch Mode You can type RMAN commands into a file, and then run the command file by specifying its name on the command line. The contents of the command file should be identical to commands entered at the command line.

When running in batch mode, RMAN reads input from a command file and writes output messages to a log file (if specified).

Recovery Manager (continued)

RMAN parses the command file in its entirety before compiling or executing any commands. Batch mode is most suitable for performing regularly scheduled backups via an operating system job control facility.

```
$ rman target / catalog rm_dbl/rm_dbl@cat @tbsbk.rcv log tbs.log
```

In this example, the user has created a file `tbsbk.rcv`, which contains the commands the user would have used interactively. RMAN would output the messages to the file `tbs.log`.

RMAN Commands

RMAN Commands

RMAN commands are of two types:

- **Stand-alone**
 - Executed individually
 - Usually do not interact with OS
 - No channel allocation
- **Job**
 - Executed as a group
 - Generally interact with OS
 - Channel allocation

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

RMAN Commands

RMAN uses two types of commands: *stand-alone* and *job* commands.

Generally stand-alone commands are self-contained. Following are some of the stand-alone commands:

- Change
- Connect
- Create Catalog, Resync Catalog
- Create Script, Delete Script, Replace Script

The job commands are usually grouped and RMAN executes the job commands inside of a *run* command block sequentially. If any command within the block fails, RMAN ceases processing—no further commands within the block are executed.

Users can execute the commands in interactive mode or batch mode. To run RMAN commands interactively, start RMAN and then type commands into the command line interface.

You can type RMAN commands into a file. You can then run a list of commands in batch mode by specifying the command file name in the command line.

Some Command Examples

To Mount the Target Database Ensure that you have connected to RMAN with the SYSDBA privilege. If you are using recovery catalog, ensure that your target database is different from the catalog database. Then issue the startup command

```
RMAN> startup mount pfile=/u01/oracle/db01/pfile/initdb01.ora
```

To Shut Down the Target Database Ensure that you have connected to RMAN with the SYSDBA privilege. Then issue the shutdown command

```
RMAN> shutdown immediate
```

To List the Current Configuration of the Target Database Use the REPORT command to get the configuration of the database.

```
RMAN> report schema;
```

```
RMAN> report schema;
```

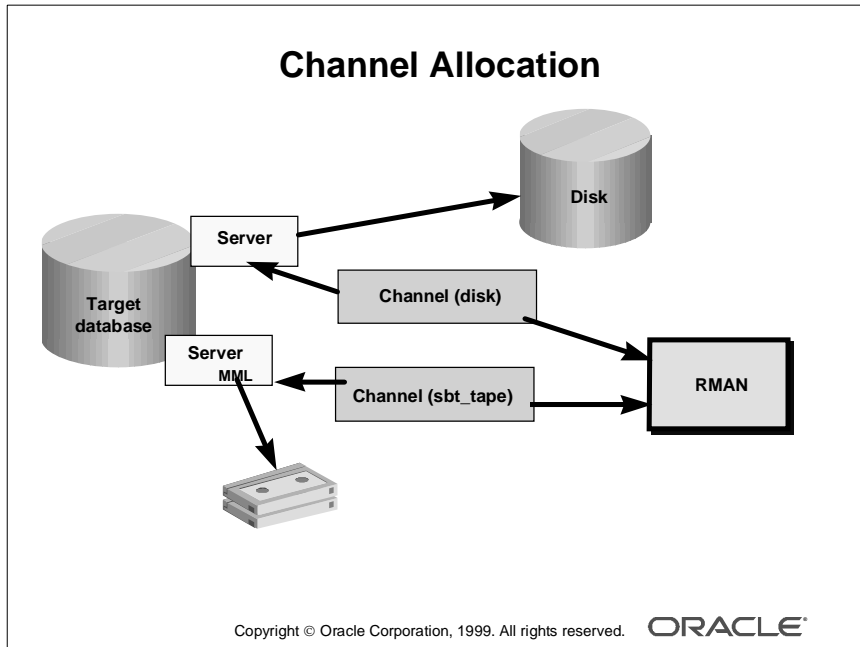
```
RMAN-03022: compiling command: report
```

```
Report of database schema
```

File	K-bytes	Tablespace	RB segs	Name
1	52704	SYSTEM	YES	/u01/db01/system.dbf
2	5120	RBS	YES	/u01/db01/rbs01.dbf
3	8192	USER_DATA	NO	/u01/db01/users0.dbf
4	1024	TEMP	NO	/u01/db01/temp01.dbf
5	1024	INDEX_DATA	NO	/u01/db01/index0.dbf
6	25600	OEM_DATA	NO	/u01/db01/oem01.dbf
7	20480	RMAN_TS	NO	/u01/db01/rman01.dbf

```
RMAN>
```

Channel Allocation



Channel Allocation


You must allocate a channel before you execute backup and recovery commands. Each allocated channel establishes a connection from RMAN to a target or auxiliary database (either a database created with the duplicate command or a temporary database used in TSPITR) instance by starting a server session on the instance. This server session performs the backup and recovery operations. Only one RMAN session communicates with the allocated server sessions.

Each channel usually corresponds to one output device, unless your media management library is capable of hardware multiplexing.

REPORT Command

REPORT Command

- Produces a detailed analysis of the recovery catalog
- Produces reports to answer:
 - Which files need a backup?
 - Which backups can be deleted?
 - Which files are unrecoverable?



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

REPORT Command

This command helps in analyzing information in the recovery catalog in more detail.

Reports can be produced for a variety of questions, such as:

- What is the structure of the database?
`RMAN> report schema;`
- Which files need to be backed up?
`RMAN> report need backup ...;`
- Which backups can be deleted (that is, are obsolete)?
`RMAN> report obsolete;`
- Which files are not recoverable because of unrecoverable operations?
`RMAN> report unrecoverable ...;`

REPORT NEED BACKUP Command

REPORT NEED BACKUP Command

- Lists all data files requiring a backup
- Assumes most recent backup used during a restore.
- Three options are available:
 - Incremental
 - Days
 - Redundancy

```
report need backup incremental 3;
```

```
report need backup days 3;
```

```
report need backup redundancy 3;
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

REPORT NEED BACKUP Command

The REPORT NEED BACKUP ... command is used to identify all data files needing a backup. The report assumes that the most recent backup would be used in the event of a restore. There are three options:

- **Incremental:** An integer specifying the maximum number of incremental backups that should be restored during recovery. If this number or more are required, then the data file needs a new full backup.
For example, report files needing three or more incrementals for recovery:

```
RMAN > report need backup incremental 3 database;
```
- **Days:** An integer specifying the maximum number of days since the last full or incremental backup of a file. The file needs a backup if the most recent backup is equal to or greater than this number.
For example, to report what system files have not been backed up for three days:

```
RMAN > report need backup days 3 tablespace system;
```
- **Redundancy:** An integer specifying the minimum level of redundancy considered necessary. For example, redundancy level two requires a backup if there are not two or more backups

LIST Command

LIST Command Operations

- Lists backup sets and copies of data files
- Lists backup sets and copies of any data file for a specified tablespace
- Lists backup sets and copies containing archive logs for a specified range
- Lists incarnations for the database

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE[®]

LIST Command

The LIST command is used to produce a detailed report listing all information for the following:

- Backup sets that contain a backup of a specified list of data files
- Copies of a specified list of data files
- Backup sets that contain a backup of any data file that is a member of a specified list of tablespaces
- Copies of any data file that is a member of a specified list of tablespaces
- Backup sets that contain a backup of any archived logs with a specified name or range
- Copies of any archived logs with a specified name or range
- Incarnations of a specified database

LIST Command

- List all copies of data files in the SYSTEM

```
RMAN > list copy of tablespace "SYSTEM";
```

- List all backup sets containing the data file user01.dbf:

```
RMAN > list backupset of datafile  
2> "/disk1/data/user01.dbf";
```

- List incarnation for database DB00:

```
RMAN > list incarnation of database DB00;
```

Copyright ©Oracle Corporation, 1999. All rights reserved.

ORACLE®

Using the LIST Command

Before using the LIST command, you must connect to RMAN specifying both the target database and the recovery catalog:

```
RMAN> target sys/oracle@DB00" \  
2> rcvcat rman/rman@RCVCAT"
```

LIST Data File Copies

The example uses the List command to list data file copies for the SYSTEM tablespace.

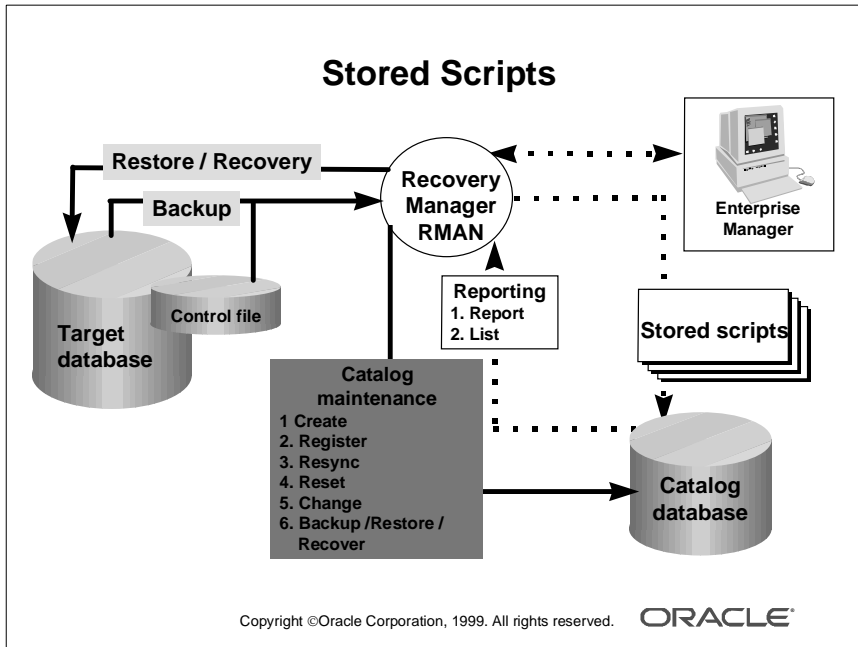
LIST Backup Set

The example uses the LIST command to list all known backups of the data file "/disk1/data/user01.dbf".

List Incarnation

The example lists all known incarnations of the database DB00.

Stored Scripts



Stored Scripts

A recovery manager script is a set of commands that:

- Specify frequently used backup, recover, and restore operations
- Are created using the `CREATE SCRIPT` command
- Are stored in the recovery catalog
- Can be called only within the `RUN` command
- Enables you to plan, develop, and test a set of commands for backing up, restoring, and recovering the database
- Minimize the potential for operator errors.

Storing Scripts

An incremental level 0 backup can be created and stored in a single script called `level0backup`. Storing the script in the recovery catalog enables any DBA using Recovery Manager to access the scripts. To find which scripts are stored, query the `RC_STORED_SCRIPT` and `RC_STORED_SCRIPT_LINE` views.

Script Examples

- **Create script**

```

RMAN> create script Level0Backup {
    allocate channel d1 type disk;
    backup
    incremental level 0
    format '/u01/db01/backup/%d_%s_%p'
    fileperset 5
    (database include current controlfile);
    sql 'alter database archive log current';
    release channel d1;}

```

- **Execute script**

```

RMAN > run {execute script Level0Backup;}

```

Copyright ©Oracle Corporation, 1999. All rights reserved.

ORACLE®

Create and Use Stored Scripts

Backup, restore and recovery operations are generally automated using scripts. RMAN provides a way of storing these scripts in the recovery catalog. You create scripts using the CREATE SCRIPT command. You have to use the RUN command to execute the script. If you need to change the script, you can use the REPLACE SCRIPT command.

Create Script Example

```

RMAN> create script level0backup {
    2> allocate channel d1 type disk;
    3> backup incremental level 0
    4> format '/u01/db01/backup/df%s_%p'
    5> filesper set 5
    6> (database include current controlfile);
    7> release channel d1; }
RMAN-03022: compiling command: create script
RMAN-03023: executing command: create script
RMAN-08085: created script level0backup
RMAN>

```

Create and Use Stored Scripts (continued)

Create Script Example (continued)

```
RMAN> run { execute script level0backup;};
RMAN-03021: executing script: level0backup
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: d1
RMAN-08500: channel d1: sid=14 devtype=DISK
RMAN-03022: compiling command: backup
RMAN-03025: performing implicit partial resync of recovery catalog
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete
RMAN-03023: executing command: backup
RMAN-08008: channel d1: starting incremental level 0 datafile
backupset
RMAN-08502: set_count=1 set_stamp=364571066 creation_time=04-MAY-
99
RMAN-08010: channel d1: specifying datafile(s) in backupset
RMAN-08522: input datafile fno=00001 name=/u01/db01/system01.dbf
RMAN-08011: including current controlfile in backupset
RMAN-08522: input datafile fno=00004 name=/u01/db01/temp01.dbf
RMAN-08522: input datafile fno=00002 name=/u01/db01/rbs01.dbf
RMAN-08013: channel d1: piece 1 created
RMAN-08503: piece handle=/u01/db01/backup/df1_1 comment=NONE
RMAN-08525: backup set complete, elapsed time: 00:00:36
RMAN-08008: channel d1: starting incremental level 0 datafile
backupset
RMAN-08502: set_count=2 set_stamp=364571103 creation_time=04-MAY-
99
RMAN-08010: channel d1: specifying datafile(s) in backupset
RMAN-08522: input datafile fno=00006 name=/u01/db01/oem01.dbf
RMAN-08522: input datafile fno=00007 name=/u01/db01/rman01.dbf
RMAN-08522: input datafile fno=00003 name=/u01/db01/users01.dbf
RMAN-08522: input datafile fno=00005 name=/u01/db01/index01.dbf
RMAN-08013: channel d1: piece 1 created
RMAN-08503: piece handle=/u01/db01/backup/df2_1 comment=NONE
```

Create and Use Stored Scripts (continued)

Create Script Example (continued)

```
RMAN-08525: backup set complete, elapsed time: 00:00:25
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete
RMAN-03022: compiling command: release
RMAN-03023: executing command: release
RMAN-08031: released channel: dl
```

Log in to operating system and determine if the files exist in the backup directory.

```
ls -l /u01/db01/backup/
total 105504
-rw-r----- 1 o815 dba 51204608 May  4 13:45 df1_1
-rw-r----- 1 o815 dba 4237824 May  4 13:45 df2_1
```

RUN Command

RUN Command Examples

- Compiles commands into blocks of PL/SQL, called steps
- Steps executed immediately from memory
- Steps can be stored in script
- To run an operating system command:

```
RMAN > run { host "ls -l"; }
```

- To run a SQL command:

```
RMAN > run { sql "alter system switch logfile"; }
```

- To run a stored script:

```
RMAN > run { execute script NightlyBackup; }
```

Copyright ©Oracle Corporation, 1999. All rights reserved.

ORACLE

Using the RUN Command

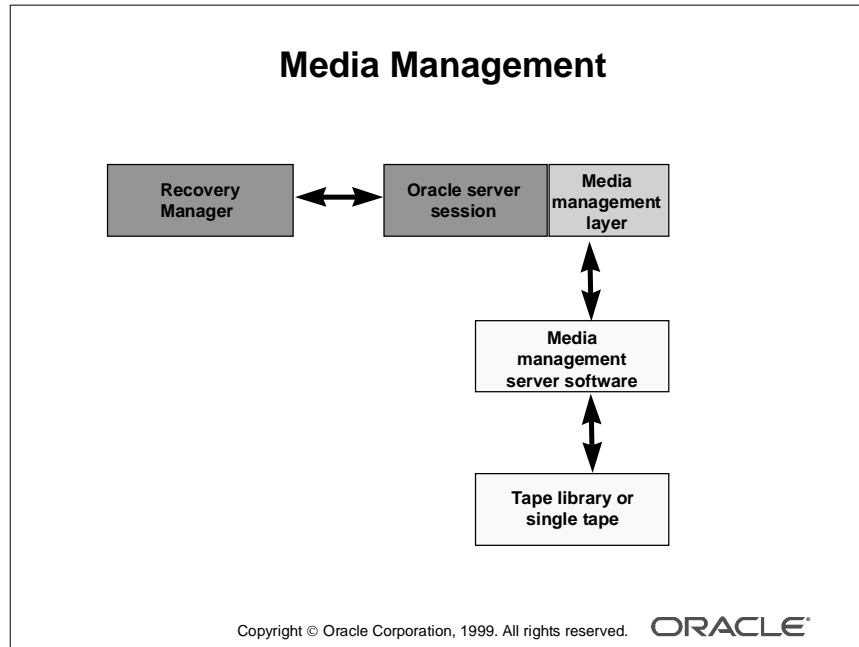
You can use the RUN command to execute a stored script. You can also run SQL and operating system commands within the RMAN command line interface using the RUN command.

- The RUN command compiles the commands into one or more blocks of PL/SQL code, called steps.
- The steps created by the compilation phase are held in memory and are executed immediately.
- RUN commands can be stored in a script. A recommended approach is to store the RUN commands in a file on the operating system, as well as to store them in the recovery catalog.

Example To run the Level0Backup script created previously, use the following syntax:

```
RMAN > run { execute script Level0Backup ; }
```

Media Management



Media Management

To use tape storage for your database backups, RMAN requires a media manager. A media manager is a utility that loads, labels, and unloads sequential media such as tape drives for the purpose of backing up, restoring, and recovering data.

Some media management products can manage the entire data movement between Oracle data files and the backup devices. Some products that use high-speed connections between storage and media subsystems can reduce much of the backup load from the primary database server.

Note that the Oracle server does not need to connect to the media management layer (MML) software when it backs up to disk. Check with your media vendor to determine whether the version of the media software you own has been certified to be compatible with RMAN. The Oracle server calls MML software routines to back up and restore data files to and from media controlled by the media manager.

Backup and Restore Operations Using a Media Manager

The following Recovery Manager script performs a data file backup to a tape drive controlled by a media manager:

```
run {  
    # Allocating a channel of type 'sbt_tape' for serial device  
    allocate channel ch1 type 'sbt_tape';  
    backup datafile 10;  
}
```

When Recovery Manager executes this command, it sends the backup request to the Oracle server session performing the backup. The Oracle server session identifies the output channel as a media management device and requests the media manager to load a tape and write the output.

The media manager labels and keeps track of the tape and names of files on each tape.

The media manager handles restore as well as backup operations. When you restore a file, the following steps occur:

- 1 The Oracle server requests the restore of a particular file.
- 2 The media manager identifies the tape containing the file and reads the tape.
- 3 The media manager passes the information back to the Oracle server session.
- 4 The Oracle session writes the file to disk.

Summary

Summary

In this lesson, you should have learned that:

- **Recovery Manager simplifies the backup, restore, and recovery process**
- **RMAN stores frequently used scripts**
- **You can use RMAN with or without recovery catalog**
- **Using recovery catalog provides RMAN with more functionality**
- **RMAN implements reporting through LIST and REPORT commands**
- **Media management layer is used with tape**

Copyright ©Oracle Corporation, 1999. All rights reserved. **ORACLE®**

Quick Reference

Context	Reference
Parameters	CONTROL_FILE_RECORD_KEEP_TIME
Dynamic performance views	
Data dictionary packages	DBMS_RCVCAT DBMS_RCVMAN DBMS_BACKUP_RESTORE
Commands	ALTER DATABASE ADD LOGFILE ALTER SYSTEM SWITCH LOGFILE

Oracle Recovery Catalog Creation and Maintenance

Objectives

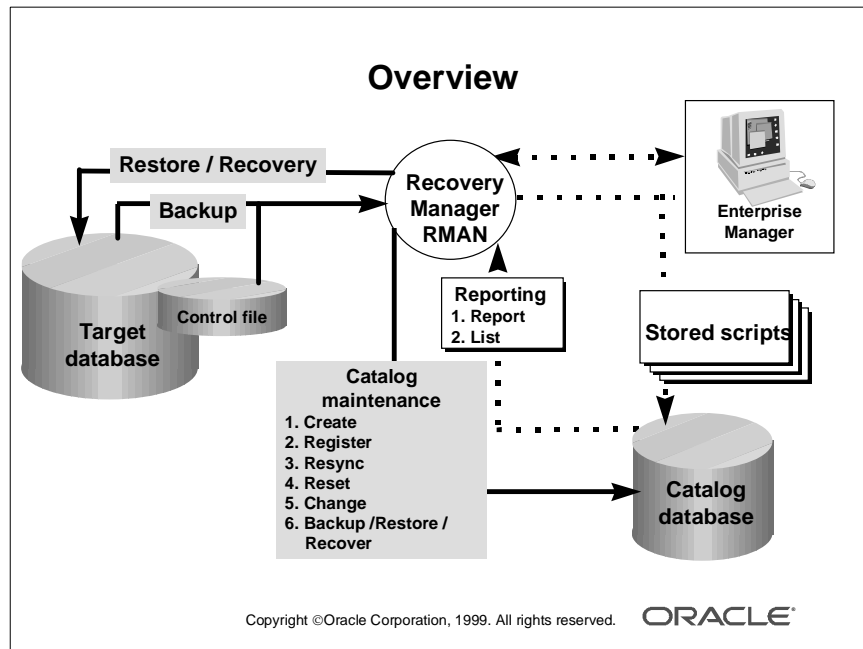
Objectives

After completing this lesson, you should be able to do the following:

- **Describe the considerations for using a recovery catalog**
- **Describe the components of a recovery catalog**
- **Create a recovery catalog**
- **Maintain the recovery catalog using recovery manager commands**
- **Query the recovery catalog to generate reports and lists**
- **Create, store, and run scripts**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

Overview



Overview

Recovery Catalog is an important component of Recovery Manager (RMAN). Even though it is not mandatory, it is recommended that you use the recovery catalog in your backup and recovery operations using the RMAN. The database in which the recovery catalog resides is referred to as the catalog database.

The Recovery Manager is used to store, use, and maintain the information in the recovery catalog. The recovery catalog is maintained using RMAN in the following operations:

- 1 Creating the catalog
- 2 Registering the catalog with the target database
- 3 Resynchronizing with the control file of the target database
- 4 Resetting the database to a previous incarnation
- 5 Changing some information about the backup/recovery files
- 6 Performing a backup, restore or recovery operation

You can use the REPORT and LIST commands of RMAN to obtain information from the recovery catalog. You could also store the backup scripts often used in the recovery catalog as stored scripts.

Recovery Catalog Considerations

- **Locate recovery catalog on a database different from the target database.**
- **Store the recovery catalog in a database on separate disks.**
- **Consider a database for storing recovery catalog for many target databases.**
- **Backup strategy should include a method for back up of the recovery catalog.**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Recovery Catalog Considerations

Information about backups, archived logs, and the structure of the target database are placed in the recovery catalog *only* by Recovery Manager.

Storage To increase the speed and ease of recovery, while simultaneously reducing the risk of data loss, use the following guidelines:

- Do not create the recovery catalog in the target database. Choose a separate database with data files on different disks.
- If the recovery catalog is located on the same system as the target database, store the recovery catalog in a database on separate disks.
- If there are many databases to back up, consider creating a separate recovery catalog database to hold the information on the target databases.
- Your backup strategy should include a method for backing up the recovery catalog.

Recovery Catalog Considerations

- **Wider variety of automated backup and recovery functions such as:**
 - **Stored scripts for backup and recovery operations**
 - **Tablespace point-in-time recovery using RMAN**
 - **Record of backup and recovery operations maintained for a time period suited to the user**
- **Recovery catalog size:**
 - **The number of databases monitored by the catalog**
 - **The number and size of scripts stored in the catalog**
 - **The number and frequency of generation of archive logs**
- **Backup recovery catalog**

Copyright ©Oracle Corporation, 1999. All rights reserved.

ORACLE®

Recovery Catalog Considerations (continued)

Facilities When you use a recovery catalog, RMAN can perform a wider variety of automated backup and recovery functions such as:

- Stored scripts for backup and recovery operations
- Maintain record of backup and recovery operations for a time period suited to the user

Size When you use a recovery catalog, RMAN requires that you maintain a recovery catalog schema as well as any associated space used by that schema. The size of the recovery catalog schema depends on the following factors.

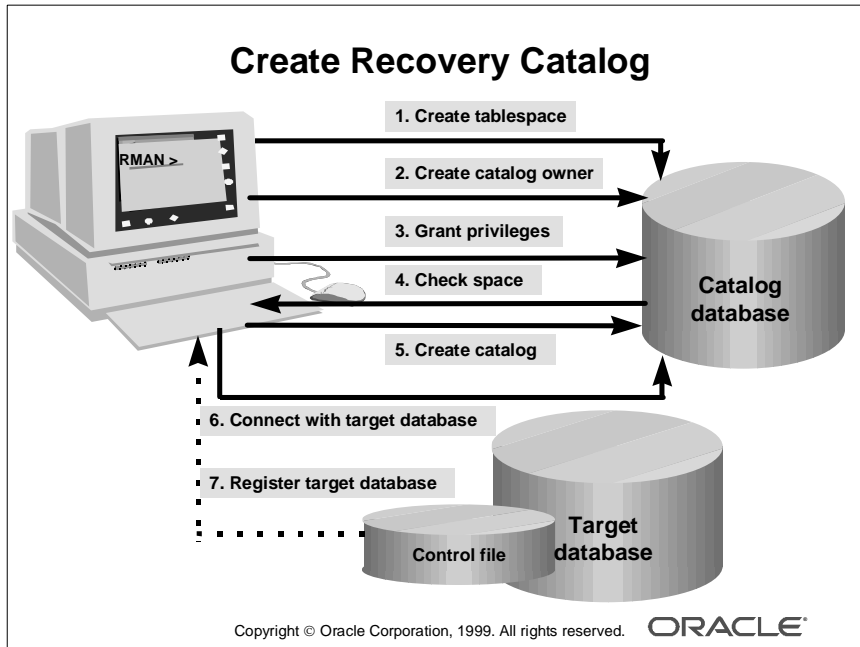
- The number of databases monitored by the catalog. It is recommended that you have a separate catalog for each database.
- The number and size of scripts stored in the catalog. Depending on the complexity and variety of backup operations being performed on the target database you may have fewer or more scripts stored in the catalog.
- The number and frequency of generation of archive logs.

Recovery Catalog Considerations (continued)

Recovery Catalog Backup If you use a recovery catalog, decide which database you will use to install the recovery catalog schema, and also how you will back up this database. If you use RMAN to back up several databases, you may want to create a separate recovery catalog database and create the RMAN user in that database. Also, it is recommended that you operate this database in ARCHIVELOG mode.

If you have more than one database to back up, you can create more than one recovery catalog and have each database serve as the other's recovery catalog. For example, assume that you maintain two production databases, one called PRD1 and a second called PRD2. You can install the recovery catalog for PRD1 in the PRD2 database, and the recovery catalog for the PRD2 database in PRD1.

Creating the Recovery Catalog



How to Create a Recovery Catalog

To create the recovery catalog, perform the following steps:

- 1 Connect to the catalog database and create a tablespace for the catalog:

```
SQL > create tablespace rman_ts datafile '<directory>/<name>'
      > size 20M default storage
      (initial 100K next 100K pctincrease 0);
```

- 2 Create a Recovery Manager user:

```
SQL > create user rman_dbl identified by rman_dbl
      2> default tablespace rman_ts
      3> temporary tablespace temp
      4> quota unlimited on rman_ts;
```

- 3 Grant the roles and privileges to this user to maintain the recovery catalog and perform the backup and recovery operations.

```
SQL > grant recovery_catalog_owner to rman_dbl;
SQL > grant connect, resource to rman_dbl;
```

- 4 Ensure that you have sufficient space in the tablespace and rollback segments. The catalog would require about 20 MB of space and the rollback segment should be about 5 M in size.

How to Create a Recovery Catalog (continued)

- 5 Log in to the operating system and run the RMAN command to invoke the RMAN command interpreter and create the catalog. Use of the MSGLOG (or LOG in Oracle8i) option enables RMAN to output messages and commands to a file.

```
% rman catalog rman_db1/rman_db1@catdb msglog = catalog.log
create catalog tablespace rman_ts;
exit;
```

Note: When you use MSGLOG the output is directed to the file and you may not get the RMAN prompt. So the CREATE CATALOG command should be entered when the cursor appears on the new line. Similarly the exit command should be entered on the next new line. The purpose of the msglog is to help you record any errors that may arise in the process of creation of catalog so that corrective actions can be taken.

- 6 Check the catalog.log file created by RMAN to note if there are any errors in creation of the catalog. If you do find any errors, you should drop all objects under the catalog owner and re-create them from the beginning.

```
Catalog.log may appear as follows:
Recovery Manager: Release 8.1.5.0.0 - Production
RMAN-06008: connected to recovery catalog database
RMAN-06428: recovery catalog is not installed
RMAN> create catalog tablespace rman_ts;
RMAN-06431: recovery catalog created
RMAN>
RMAN> exit;
Recovery Manager complete
```

- 7 Connect using the RMAN executable on the server containing the target database. You should log in as a user with SYSDBA privileges on the target database to perform all the backup and recovery operations.

```
% rman target sys/oracle@db1
Recovery Manager: Release 8.1.5.0.0 - Production
RMAN-06005: connected to target database: DB1 (DBID=472633597)
RMAN> connect catalog rman_db1/rman_db1@catdb
RMAN-06008: connected to recovery catalog database
RMAN>
```


How to Create a Recovery Catalog (continued)

- 8** Register the target database in the catalog. If the target database is not registered in the recovery catalog, the catalog may not be used to store information about the database. Recovery Manager uses the internal database identifier (DBID), which is calculated when the database is first created, as a unique identifier for the database. If you attempt to register a new database that has been created by copying an existing database and then changing the `db_name`, the register will fail. To back up a copied database, create a new recovery catalog owner, and create the catalog in the new account

```
RMAN> register database;
```

```
RMAN-03022: compiling command: register
```

```
RMAN-03023: executing command: register
```

```
RMAN-08006: database registered in recovery catalog
```

```
RMAN-03023: executing command: full resync
```

```
RMAN-08002: starting full resync of recovery catalog
```

```
RMAN-08004: full resync complete
```

```
RMAN>
```

Connecting Using a Recovery Catalog

Connecting Using a Recovery Catalog

When initiating the session on target database:

on UNIX:

```
$ ORACLE_SID=db01; export ORACLE_SID
$ rman target sys/oracle
RMAN> connect catalog rman_db1/rman_db1@catdb
```

on NT:

```
C:\> set ORACLE_SID=db01
C:\> rman target sys/oracle
RMAN> connect catalog rman_db1/rman_db1@catdb
```

Remote connection

```
rman target sys/oracle@db01
RMAN-6005: connected to target database: ...
RMAN> connect catalog rman_db1/rman_db1@catdb
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

How to Connect to Recovery Manager

To connect to RMAN using a recovery catalog, follow these steps:

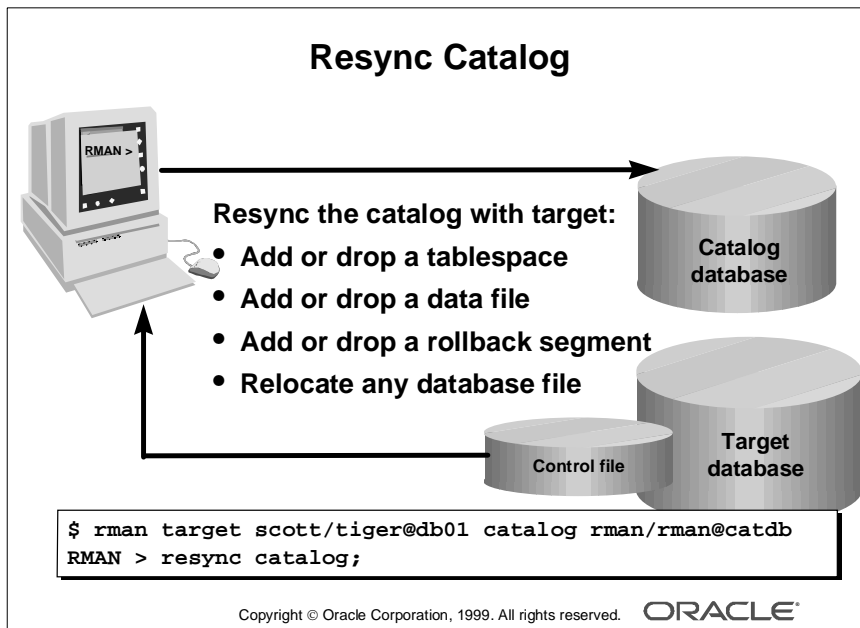
- 1 Initiate an RMAN session from the target database. Because backup and recovery operations very often require privileged operations, you may like to connect as a user with SYSDBA privilege.

On Windows NT you may invoke the command prompt before issuing the following RMAN command or you can invoke the command line interface using the Run option from Start menu.

```
rman target sys/oracle@db01
RMAN-06005: connected to target database: NEW8I(DBID=XXXXXXXXXX)
RMAN>
```

- 2 Connect to the recovery catalog. In case you have more than one recovery catalog, you should assign a catalog to a target database.

```
RMAN> connect catalog rman_db1/rman_db1@catdb
RMAN-06008: connected to recovery catalog database
RMAN>
```



Resync Catalog

Most of the information in the recovery catalog is derived from the control file of the target database. Any structural changes, such as adding tablespaces to the database, cause the control file and recovery catalog to become “out of synch” if the recovery catalog is not updated. The catalog would be synchronized any time a new BACKUP or COPY command is issued with the catalog. However, this synchronization could cause a delay in the backup operation.

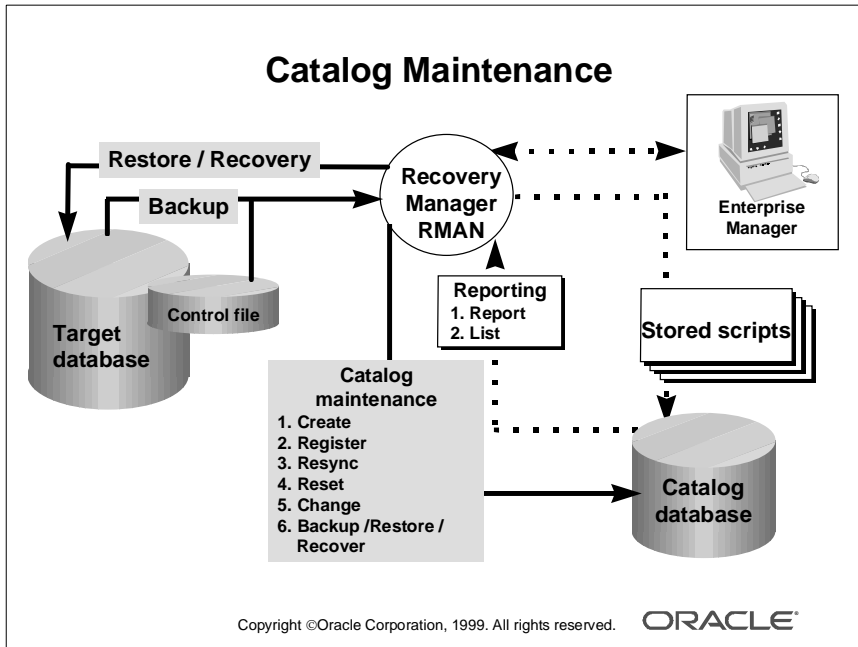
The following factors may influence how often you resync:

- The rate of archive log creation
- Frequency of database structure changes
- The need for faster resynchronizations

The RESYNC CATALOG command updates the following records:

- Log switch records: Created when a log switch occurs. Recovery Manager tracks this information so it knows what archive logs it should expect to find.
- Archived log copy records: Associated with archived logs that were created by archiving an online log, by copying an existing archived log, or by restoring an archived log backup set.
- Backup history records: Associated with backup sets, backup pieces, backup set members, and file copies. The RESYNC CATALOG command is needed only when the recovery catalog database was unavailable when a BACKUP or COPY command was executed.

Catalog Maintenance



CHANGE and CATALOG Commands

You have learned the RESYNC command to synchronize the recovery catalog with the control file of the target database. Some operations may be performed in the target database, such as backing up a tablespace using the operating system, that are not recorded in the control file of the target database. To effectively use such a backup, you should record this information in recovery catalog. CHANGE and CATALOG commands can be used to manually update the recovery catalog.

Some examples of change and catalog commands are discussed in this lesson. For detailed syntax, refer to *Oracle8i Backup and Recovery Guide*.

Catalog command

Use the CATALOG command to add information to the catalog

- When backup is taken using OS
- When backup is taken without recovery catalog

The CATALOG command stores:

- Archived logs, data file copies, and control file copies that belong to the target database
- Only files with the same database incarnation number

```
RMAN > catalog datafilecopy '/u01/db1/backup/system01.bak'  
2 > tag = 'SYSTBS0429';  
  
RMAN > catalog archivelog  
2> '/u01/db1/ARCHIVE/arch_267.log';
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Adding Information to the Catalog

If you have used the operating system to perform a backup of a tablespace, you should record this information in the recovery catalog. RMAN can then make use of this backup in case a recovery is required. This information can be recorded in the recovery catalog by using the CATALOG command.

You can use the CATALOG command to add data files copy information, archived log information, or control file copy information to the recovery catalog for:

- Files that were created before Recovery Manager was installed
- Files that were created without using RMAN (for example, OS backups)
- Only those files with the same incarnation number as the current incarnation number of the database
- Only those files with an Oracle8 or later format
- Only those files that belong to the target database

Adding Information to the Catalog (continued)

Example In this scenario a backup of system tablespace has been taken without use of RMAN. Now the existence of this backup is being registered in the recovery catalog.

```
rman target sys/change_on_install@db01
RMAN > connect catalog rman_dbl/rman_dbl@catdb
RMAN> catalog datafilecopy '/u01/db1/backup/system01.bak' tag
'systbs0429';
RMAN-03022: compiling command: catalog
RMAN-03023: executing command: catalog
RMAN-08050: cataloged datafile copy
RMAN-08513: datafile copy filename=/u01/db1/backup/system01.bak
recid=1 stamp=364062433
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete
RMAN> list backup of datafile
```

CHANGE Command

CHANGE Command

- Mark a backup piece, image copy, or archived redo log as **unavailable** or **available**.
- Delete a backup piece, image copy, or archived redo log from the operating system and update its recovery catalog record to **deleted** status.
- Verify whether backup pieces, data file copies, or archived redo logs are available and, if they are not, mark them as **expired**.
- This command operates on backup set, backup piece, data file copy, archivelog, and control files.

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

Changing Information in the Recovery Catalog

Sometimes the information stored in the recovery catalog may be outdated and not in sync with the data available on the disk or tape. The CHANGE command enables you to ensure that the recovery catalog is synchronized with the available backup information. Some options of the CHANGE command are:

- To mark a data file copy or backup set as available, use the **AVAILABLE** option.

```
CHANGE DATAFILECOPY '/u01/db01/backup/systbs.bak' available;
CHANGE BACKUPSET 12 AVAILABLE;
```

- Remove references from the recovery catalog if the physical file does not exist. For example, when an archived log was physically removed using an OS utility, use the CHANGE command to record it in the catalog:

```
change archivelog '/disk1/archive/arch_123.rdo' uncatalog;
```

- Check the presence of backup set, data file copy or archive log by using the **CROSSCHECK** option.

```
CHANGE ARCHIVELOG ALL CROSSCHECK;
CHANGE DATAFILECOPY '/u01/db01/backup/systbs.bak' CROSSCHECK;
```

If RMAN does not find the copy, you will see the message

```
RMAN-06153: validation failed for datafile copy.
```

Change - Mark Unavailable

- Record if a backup is temporarily unavailable in the recovery catalog.

```

RMAN > change datafilecopy
2> '/disk1/data/system01.bak' unavailable;
    
```

- Use the LIST command to check the status.

```

RMAN> list copy;
List of Datafile Copies
Key File S Completion time Ckp SCN    Ckp time
Name
-----
                Unavailable
-----
2096 1  (U) 05-MAY-99      435490    05-MAY-99
/u01/ db01/BACKUP/system01.bak
    
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

To Mark a Data File Copy as Unavailable

If the backup is only temporarily unavailable, then you can record this backup as unavailable in the recovery catalog. In this case the record is retained in the catalog but marked as unavailable. The third column S(tatus) in the list shows U(navailable). Because we are updating only the recovery catalog, no channels are required.

```

RMAN> change datafilecopy '/u01/db1/backup/system01.bak'
unavailable;
RMAN-03022: compiling command: change
RMAN-06108: changed datafile copy unavailable
RMAN-08513: datafile copy filename=/u01/db1/backup/system01.bak
recid=1 stamp=364062433
    
```

Use the LIST command to check if the status of the record has changed.

```

RMAN> list copy;
List of Datafile Copies
Key File S Completion time Ckp SCN    Ckp time  Name
-----
2096    1  U   05-MAY-99      435490    05-MAY-99 /u01/db01/
BACKUP/
system01.bak
    
```


Deleting a Backup

Deleting a Backup Record

To physically delete backups and copies and update their metadata records:

```
$ rman target sys/change_on_install@db01
RMAN> connect catalog rman_db1/rman_db1@catdb

RMAN > allocate channel for delete type disk;

RMAN > change datafilecopy
      2> '/u01/db01/backup/system01.bak' delete;

RMAN > release channel;
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

Deleting a Data File Copy

Remove old files from the control file, the recovery catalog, and from the media where they reside. The CHANGE . . . DELETE command requires a channel to delete a physical file from a disk. The FOR DELETE option of ALLOCATE CHANNEL command is used only for deleting files. The delete type of channel cannot be used as an input or output channel for a job.

Example

```
RMAN> allocate channel for delete type disk;
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: delete
RMAN-08500: channel delete: sid=13 devtype=DISK
RMAN> change datafilecopy '/u01/db1/backup/system01.bak' delete;
RMAN-03022: compiling command: change
RMAN-08070: deleted datafile copy
RMAN-08513: datafile copy filename=/u01/db1/backup/system01.bak
recid=1 stamp=364062433
RMAN> release channel;
RMAN-03022: compiling command: release
RMAN-03023: executing command: release
RMAN-08031: released channel: delete
```

Deleting a Data File Copy (continued)

Example (continued)

Check if the file exists on the disk. On UNIX, use the `ls` command with the appropriate command option. On NT, you could use the explorer or DIR command (at the command prompt) to determine if the file exists.

```
RMAN> exit
% ls -l /u01/db1/backup/system01.bak
/u01/db1/backup/system01.bak: No such file or directory
```

Delete Any Backup That Has Been Marked Expired RMAN does not delete any backup pieces that it is unable to find, but updates their metadata records to expired status. You can use the `DELETE` option of the `CHANGE` command to update expired backup records in the catalog to status deleted (and delete any existing expired backup pieces).

```
RMAN> allocate channel for delete type disk;
RMAN> delete expired backup;
RMAN> release channel;
```

Remove Record from Catalog You can use the `UNCATALOG` option of the `CHANGE` command to remove backup records from the catalog without actually deleting the files from the medium. Because the file is not being acted upon by RMAN, this command option does not require a channel to be allocated.

```
RMAN> change controlfilecopy '/u01/db01/backup/sysbak'
2>      uncatalog;
RMAN> change datafilecopy 4833 uncatalog;
```

Deleting Records of Previous Incarnation

Delete Records of a Previous Incarnation

- **Allocate a channel of type maintenance.**
`allocate channel for maintenance type 'sbt_tape';`
- **Change the records to deleted status.**
`change backupset 100, 101, 102, 103 delete;`
`release channel;`
- **Connect to the catalog database in SQL*PLUS session as catalog owner and obtain DBINC_KEY.**
`$ sqlplus rman_db1/rman_db1@catdb`
`SQL> SELECT * FROM rc_database_incarnation;`
- **Delete records from dbinc.**
`SQL> DELETE FROM dbinc WHERE dbinc_key=key_value;`
- **RMAN will remove the specified incarnation records from the recovery catalog.**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

How to Delete Records of a Previous Incarnation

- 1 Allocate a channel of type maintenance:
`allocate channel for maintenance type 'sbt_tape';`
- 2 Query the catalog or the views to obtain primary keys for records referencing unwanted backup pieces, archived redo logs, and image copies. Change them to deleted status using the CHANGE ... DELETE command. Then release the channel.
`change backupset 100, 101, 102, 103 delete;`
`release channel;`
- 3 Connect to the catalog database in SQL*PLUS session as catalog owner.
`$ sqlplus rman_db1/rman_db1@catdb`
- 4 Obtain the DBINC_KEY values for the incarnation whose records you want to delete by querying the RC_DATABASE_INCARNATION view:
`SQL> SELECT * FROM rc_database_incarnation;`
- 5 Execute the following DML statement, where key_value is the value of DBINC_KEY:
`SQL> DELETE FROM dbinc WHERE dbinc_key=key_value;`
- 6 RMAN will remove the specified incarnation records from the recovery catalog.

Recovery Catalog Backup

Backup of Recovery Catalog

- **Export:**
 - If catalog database is not very large, you can export the database at regular intervals.
 - If catalog database is large, export the schema containing the recovery catalog.
- **Tablespace backup:**

If recovery catalog is stored in a separate tablespace, the database is operated in ARCHIVELOG mode
- **Database backup:**

You could take a whole database backup using the operating system or RMAN.
- **Locate recovery catalog in a database different from target database**

Copyright © Oracle Corporation, 1999. All rights reserved.

ORACLE

Backup of Recovery Catalog

It is critical to have a tested backup strategy for the recovery catalog. The recovery catalog is a schema of objects stored in a database. The considerations for backup of the recovery catalog are similar to those of a schema. You should ensure that each catalog schema is dedicated for one database only.

You could use one of the following strategies to back up the recovery catalog:

- **Export:** If the database containing the catalog is not very large, you can export the database at regular intervals. However, when the catalog database is quite large, export may take very long time and consume a large disk storage. Then you can export the schema containing the recovery catalog.
- **Tablespace backup:** If the recovery catalog is stored in a separate tablespace (as recommended) and the catalog database is operated in ARCHIVELOG mode, you can take an online backup of the tablespace containing the recovery catalog.
- **Database Backup:** You could take a whole database backup using the operating system or RMAN.

Always store the recovery catalog in a database different from your target database. Also ensure that the files related to catalog database are isolated on disks different from those containing the target database.

Recovering the Recovery Catalog

Recovering Recovery Catalog

- **Create a database from previous backup of the recovery catalog database.**
- **Relocate the catalog in another database and import the data into the new schema from export dump of the previous catalog schema.**
- **Import the entire database export dump from the recovery catalog.**

Resync the catalog immediately after the recovery catalog has been rebuilt.

Remove any unwanted records by issuing the `CHANGE . . . UNCATALOG` command.

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

Recovering the Recovery Catalog

The strategy to recover a recovery catalog would depend on the nature of failure and the backup strategy in place.

If the database containing the recovery catalog is damaged, and has to be rebuilt, then you could consider the following recovery options:

- You can create a database from previous backup of the recovery catalog database.
- You can decide to locate the catalog in another database. In that database, create a user and grant the user the `RECOVERY_CATALOG_OWNER` privilege. You can import the data from the export dump of the previous catalog owner in to the schema of the newly created user.
- You can create a new database and import the entire database export dump from the recovery catalog.
- When the recovery catalog has been rebuilt, you should resync the catalog with the control file of the target database immediately.
- During resync, Recovery Manager may add records for files that no longer exist, because files being recataloged are not verified. Remove such records by issuing the `CHANGE . . . UNCATALOG` command.

Reports

- **Reporting data files needing backup:**

```
report need backup days 5 tablespace
system;

report need backup redundancy 2 datafile
1,2,3 ;
```

- **Reporting unrecoverable data files:**

```
report unrecoverable database device type
'sbt_tape';
```

- **Reporting obsolete backups and copies:**

```
report obsolete redundancy 3 device type
disk;
```

Copyright ©Oracle Corporation, 1999. All rights reserved.

ORACLE®

Reporting Data Files Needing Backup

The following command reports all data files from tablespace SYSTEM that have not had a backup (full or incremental) in five or more days:

```
report need backup days 5 tablespace system;
```

The following command reports which of data files 1–3 need backups that do not have two or more backups or copies:

```
report need backup redundancy 2 datafile 1,2,3 ;
```

Reporting Unrecoverable Data Files

The following example reports on all data files on tape that need a new backup because they contain unlogged changes that were made after the last full or incremental backup:

```
report unrecoverable database device type 'sbt_tape';
```

Reporting Obsolete Backups and Copies

The following command reports all backups and copies on disk that are obsolete because three more recent backups or copies are already available:

```
report obsolete redundancy 3 device type disk;
```

LIST Command

LIST Command

- **Backups of data files in tablespace TBS_1 that were made since May 1, 1999:**
`list backup of tablespace tbs_1 completed before 'May 1 1999 00:00:00';`
- **Backup sets on serial devices:**
`list backup of database device type 'sbt_tape';`
- **Copies of data file 2 using the tag weekly_df2_copy that are in the copy subdirectory:**
`list copy of datafile 2 tag weekly_df2_copy like '/copy/%';`

Copyright ©Oracle Corporation, 1999. All rights reserved.

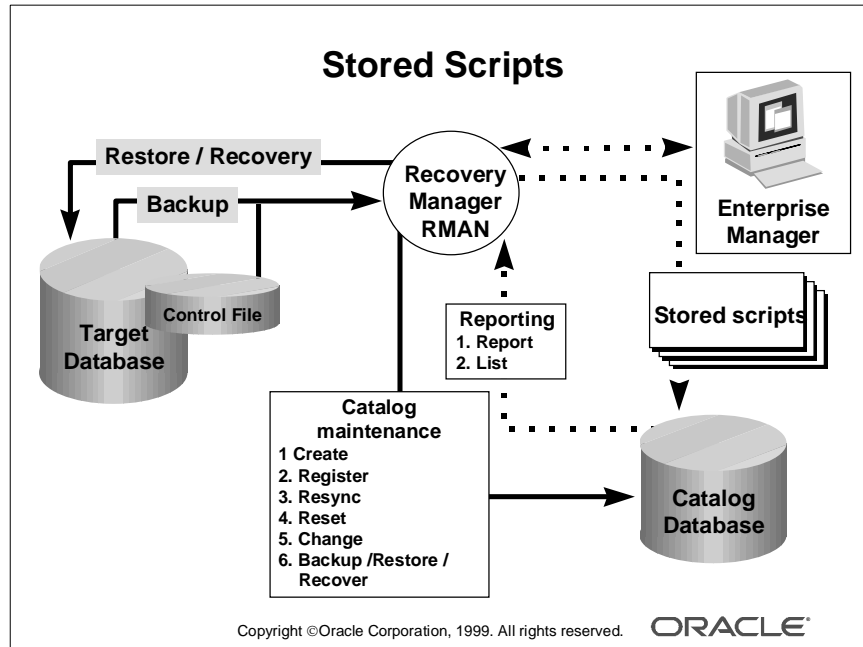
ORACLE

Making Lists of Backups and Copies

Use the LIST command to query the contents of the recovery catalog or the target database control file if no recovery catalog is used. You can use several different parameters to qualify your lists.

- The following example lists all backups of data files in tablespace TBS_1 that were made since May 1, 1999:
`list backup of tablespace tbs_1 completed before 'May 1 1999 00:00:00';`
- The following example lists all backup sets on media management devices:
`list backup of database device type 'sbt_tape';`
- The following example lists all copies of data file 2, using the tag weekly_df2__copy, that are in the copy sub-directory:
`list copy of datafile 2 tag weekly_df2_copy like '/copy/%';`
- The following example lists database incarnations registered in the recovery catalog:
`list incarnation of database;`

Stored Scripts



Stored Scripts

A Recovery Manager script is a set of commands that:

- Specify frequently used backup, recover, and restore operations
- Are created using the `CREATE SCRIPT` command
- Are stored in the recovery catalog
- Can be called only by using the `RUN` command
- Enable you to plan, develop, and test a set of commands for backing up, restoring, and recovering the database
- Minimize the potential for operator errors

Storing Scripts

An incremental level 0 backup can be created and stored in a single script called `level0backup`. Storing the script in the recovery catalog enables any DBA using Recovery Manager to access the scripts. To find which scripts are stored, query the `RC_STORED_SCRIPT` and `RC_STORED_SCRIPT_LINE` views.

Script Examples

- **Create script**

```
RMAN> create script Level0Backup {
    allocate channel d1 type disk;
    backup
    incremental level 0
    format '/u01/db01/backup/%d_%s_%p'
    fileperset 5
    (database include current controlfile);
    sql 'alter database archive log current';
    release channel d1;}
```

- **Execute script**

```
RMAN > run {execute script Level0Backup;}
```

Copyright © Oracle Corporation, 1999. All rights reserved.

ORACLE®

Creating and Using Stored Scripts

Backup, restore, and recovery operations are generally automated, using scripts. RMAN provides a way of storing these scripts in the recovery catalog. You create scripts by using the CREATE SCRIPT command. You have to use the RUN command to execute the script. If you need to change the script, you can use the REPLACE SCRIPT command.

Create Script Example

```
RMAN> create script level0backup {
  2> allocate channel d1 type disk;
  3> backup incremental level 0
  4> format '/u01/db01/backup/df%s_%p'
  5> fileperset 5
  6> (database include current controlfile);
  7> release channel d1; }
RMAN-03022: compiling command: create script
RMAN-03023: executing command: create script
RMAN-08085: created script level0backup
RMAN>
```

Creating and Using Stored Scripts (continued)

Execute Script Example

```
RMAN> run { execute script level0backup;};
RMAN-03021: executing script: level0backup
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: d1
RMAN-08500: channel d1: sid=14 devtype=DISK
RMAN-03022: compiling command: backup
RMAN-03025: performing implicit partial resync of recovery catalog
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete
RMAN-03023: executing command: backup
RMAN-08008: channel d1: starting incremental level 0 datafile
backupset
RMAN-08502: set_count=1 set_stamp=364571066 creation_time=04-MAY-
99
RMAN-08010: channel d1: specifying datafile(s) in backupset
RMAN-08522: input datafile fno=00001 name=/u01/db01/system01.dbf
RMAN-08011: including current controlfile in backupset
RMAN-08522: input datafile fno=00004 name=/u01/db01/temp01.dbf
RMAN-08522: input datafile fno=00002 name=/u01/db01/rbs01.dbf
RMAN-08013: channel d1: piece 1 created
RMAN-08503: piece handle=/u01/db01/backup/df1_1 comment=NONE
RMAN-08525: backup set complete, elapsed time: 00:00:36
RMAN-08008: channel d1: starting incremental level 0 datafile
backupset
RMAN-08502: set_count=2 set_stamp=364571103 creation_time=04-MAY-
99
RMAN-08010: channel d1: specifying datafile(s) in backupset
RMAN-08522: input datafile fno=00006 name=/u01/db01/oem01.dbf
RMAN-08522: input datafile fno=00007 name=/u01/db01/rman01.dbf
RMAN-08522: input datafile fno=00003 name=/u01/db01/users01.dbf
RMAN-08522: input datafile fno=00005 name=/u01/db01/index01.dbf
RMAN-08013: channel d1: piece 1 created
RMAN-08503: piece handle=/u01/db01/backup/df2_1 comment=NONE
RMAN-08525: backup set complete, elapsed time: 00:00:25
```

Creating and Using Stored Scripts (continued)

Execute Script Example (continued)

```
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete
RMAN-03022: compiling command: release
RMAN-03023: executing command: release
RMAN-08031: released channel: dl
```

Log in to the operating system and determine whether the files exist in the backup directory.

```
ls -l /u01/db01/backup/
total 105504
-rw-r----- 1 o815 dba 51204608 May  4 13:45 df1_1
-rw-r----- 1 o815 dba 4237824 May  4 13:45 df2_1
```

RESET DATABASE Command

RESET DATABASE Command

A database is said to be in a new incarnation when:

- The database is recovered to point-in-time.
- The database is opened with the RESETLOGS option.
- RMAN cannot use the recovery catalog again until a RESET DATABASE command is issued.
- Distinguish between a RESETLOGS and an accidental restore of an old control file.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE[®]

Using the RESET DATABASE Command

If the target database is recovered to a point in the past, the database must be opened with the RESETLOGS option, which is known as incomplete recovery. In this case, Recovery Manager cannot use the recovery catalog again until a RESET DATABASE command is issued. This assists Recovery Manager in distinguishing between a RESETLOGS and an accidental restore of an old control file.

Example

```
RMAN> reset database;
RMAN-03022: compiling command: reset
RMAN-03023: executing command: reset
RMAN-08006: database registered in recovery catalog
RMAN-03023: executing command: full resync
RMAN-08002: starting full resync of recovery catalog
RMAN-08004: full resync complete
```

Incarnation Number

An incarnation of a database is a number used to identify a version of the database prior to the log sequence number being reset to zero. This prevents archived and online redo logs from being applied to an incorrect incarnation of the database. The RESET DATABASE command is used by Recovery Manager to store database incarnation information in the recovery catalog. All subsequent backups and log archives done by the target database are associated with the new database incarnation.

RESET DATABASE TO INCARNATION Command

The RESET DATABASE TO INCARNATION <identifier> command is used to undo the effects of a RESETLOGS operation by restoring backups of a prior incarnation of the database. You must specify the primary key of the record for the database incarnation to which you return:

```
RMAN> reset database to incarnation <identifier>;
```

Note: The identifier is obtained by the LIST INCARNATION OF DATABASE command.

Example

```
RMAN> list incarnation of database;
RMAN-03022: compiling command: list
RMAN-06240: List of Database Incarnations
RMAN-06241: DB Key    Inc Key    DB Name      DB ID      CUR  Reset SCN
RMAN-06242: -----  -
RMAN-06243:         1      2      ORACLE  1186311932  YES      25730
RMAN-06243:         1    421      ORACLE  1186311932  NO       172279
RMAN> reset database to incarnation 421;
```

Data Dictionary Views

Viewing the Recovery Catalog

Data dictionary views:

- RC_DATABASE
- RC_TABLESPACE
- RC_DATAFILE
- RC_STORED_SCRIPT
- RC_STORED_SCRIPT_LINE



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

Data Dictionary Views

When the recovery catalog is created, many data dictionary views are created for accessing information from the recovery catalog. Some of the important views are used in examples here:

Example 1

To determine which databases are currently registered in the recovery catalog:

```
SQL> select * from rc_database;
```

DB_KEY	DBINC_KEY	DBID	NAME	CHANGE#	RESETLOGS
1	2	1943591421	DB01	1	20-APR-99

Data Dictionary Views (continued)

Example 2

To determine which tablespaces are currently stored in the recovery catalog for the target database:

```
SQL > select DB_KEY, DBINC_KEY, DB_NAME, TS#, NAME,
           CREATION_CHANGE# CHANGE#, CREATION_TIME CRE_DATE
       from rc_tablespace;
```

DB_KEY	DBINC_KEY	DB_NAME	TS#	NAME	CHANGE#	CRE_DATE
-----	-----	-----	--	-----	-----	-----
1	2	DB01	3	DATA01	9611	20-APR-99
1	2	DB01	1	RBS	9599	20-APR-99
1	2	DB01	4	RMAN_TS	14023	29-APR-99
1	2	DB01	0	SYSTEM	3	20-APR-99
1	2	DB01	2	TEMP	9605	20-APR-99

Example 3

To determine which scripts are currently stored in the recovery catalog for the target database:

```
SQL> select * from rc_stored_script;
```

DB_KEY	DB_NAME	SCRIPT_NAME
-----	-----	-----
1	DB01	nightlybackup
1	DB01	archivebackup

Summary

Summary

In this lesson, you should have learned that:

- Before using the recovery catalog, you must register the target database.
- You should resynchronize the catalog frequently using the control file.
- The **CHANGE** and **CATALOG** commands manually update the catalog.
- The **REPORT** command analyzes the catalog.

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

Summary

- The LIST command lists information contained in the catalog.
- Recovery Manager can store scripts in the recovery catalog.
- Use the RUN command to execute scripts.
- Query information from the recovery catalog through data dictionary views.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Quick Reference

Context	Reference
Parameters	None
Dynamic performance views	None
Data dictionary views	RC_ARCHIVED_LOG RC_BACKUP_CONTROLFILE RC_BACKUP_CORRUPTION RC_BACKUP_DATAFILE RC_BACKUP_PIECE RC_BACKUP_SET RC_CONTROLFILE_COPY RC_COPY_CORRUPTION RC_DATABASE RC_DATABASE_INCARNATION RC_DATAFILE RC_DATAFILE_COPY RC_STORED_SCRIPT RC_STORED_SCRIPT_LINE
Commands	REGISTER DATABASE RESET DATABASE RESYNC CATALOG CHANGE CATALOG REPORT LIST CREATE SCRIPT { <commands>; } REPLACE SCRIPT { <commands>; } DELETE SCRIPT <name> PRINT SCRIPT <name> RUN

Backups Using Recovery Manager

Objectives

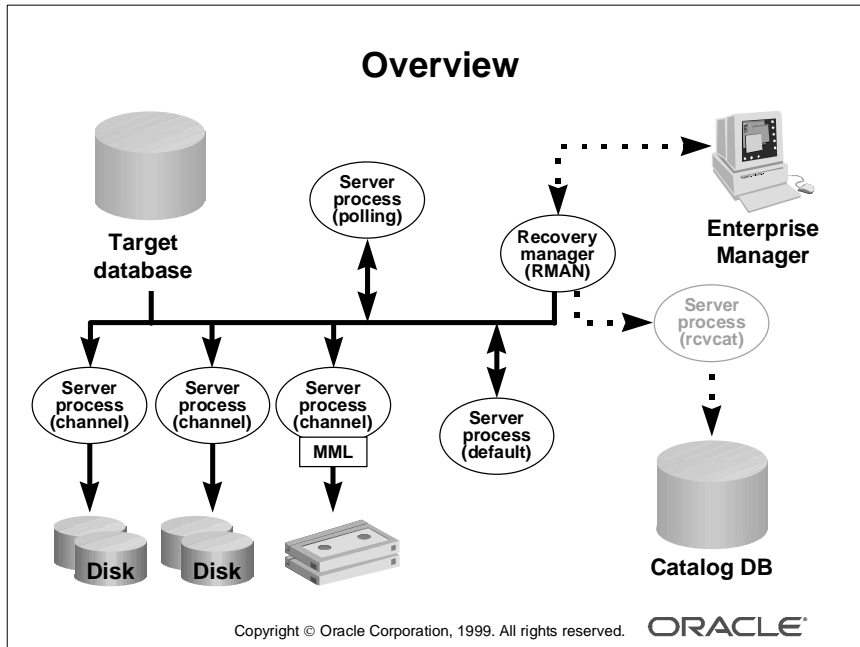
Objectives

After completing this lesson, you should be able to do the following:

- **Describe backup concepts using Recovery Manager**
- **Describe types of Recovery Manager backups**
- **Perform incremental and cumulative backups**
- **Tune backup operations**
- **View information from the data dictionary**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

Overview



Overview

Recovery Manager uses Oracle server processes to perform backup, restoration, and recovery operations, so the backup operation using Recovery Manager is called a server-managed backup.

The frequency of backup depends on the business requirements and the resources available.

Backup Concepts

Backup Concepts

- **Recovery Manager backup is a server-managed backup**
 - Recovery Manager uses Oracle server processes for backup operations
 - Includes database, tablespaces, data files, control files, archive logs
- **Closed backup**
 - Target database must be mounted (not open)
 - Includes data files, control files
- **Open Backup**
 - Tablespaces should not be put in backup mode
 - Includes data files, control files, archive logs

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

Types of Recovery Manager Backups

Recovery Manager provides functionality to back up:

- The entire database, every data file in a tablespace, or a single data file
- The control file
- All or selected archived logs

Note: The online redo log files are not backed up when using Recovery Manager.

Closed Database Backups

This is defined as a backup of the database while it is closed (offline). This is the same as the consistent database backup. If you are performing a closed backup, the target database must not be open. If you are using a recovery catalog, the recovery catalog database must also be open.

Open Database Backups

This is defined as a backup of any portion of the database while it is open (online). When using Recovery Manager, do not put tablespaces in backup mode using the ALTER TABLESPACE ... BEGIN BACKUP command. Recovery Manager uses server processes to make copies of data files, control files, or archive logs and generates less redo.

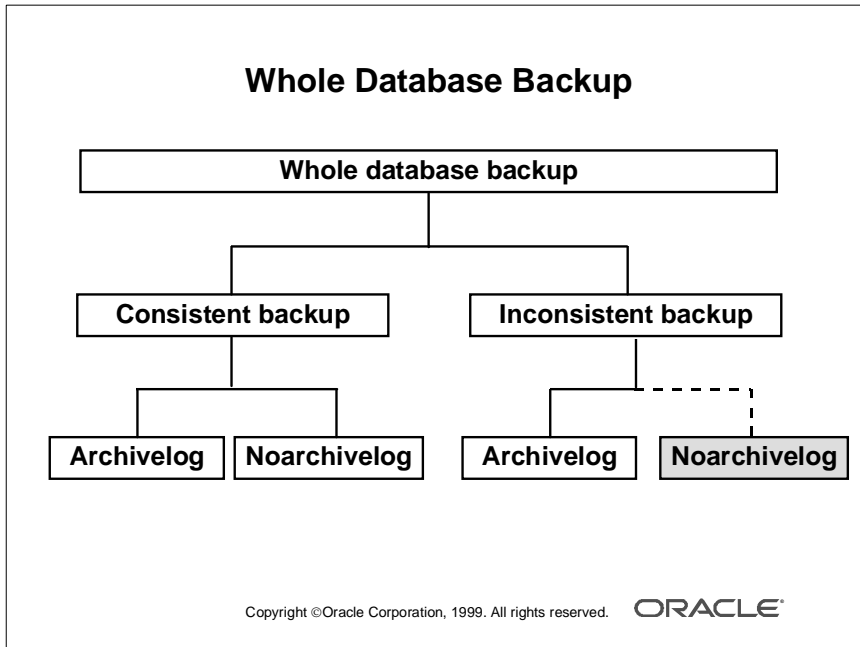
Full Backups

A full backup differs from a whole database backup. A whole backup comprises all of the data files and control file of the target database, whereas a full backup may contain one or more of the data files, control file or archive logs.

When performing a full backup, an Oracle server process reads the entire file and copies all blocks into the backup set, skipping only data file blocks that have never been used. The server session does not skip blocks when backing up archived redo logs or control files.

Full backup is not a part of the incremental backups. You can create and restore full backups of data files, data file copies, tablespaces, database, control files, archive logs and archive log copies. Note that backup sets containing archived redo logs are always full backups.

Whole Database Backup



Terminology

- **Whole backup:**
 - Target database may be open or closed
 - Backup of all data files and control file
- **Full backup:** Backup of one or more files, which is not incremental
- **Incremental backup:** Backup of data files changed since the last incremental backup
- **Image copy:** Copy of a data file, control file, or archive log on to disk

Copyright © Oracle Corporation, 1999. All rights reserved.

ORACLE®

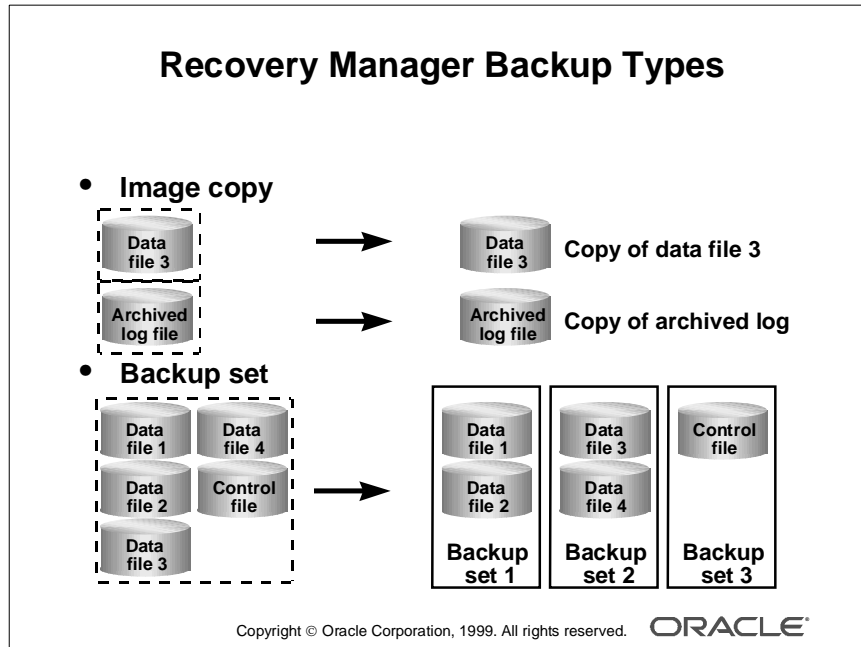
Whole Database Backup

Whole database backup (also known as whole backup) refers to a backup of all data files and the control file of the database. Whole backups can be performed when the database is closed or open. This is the most common method of backup.

The whole backup taken when the database is closed (after the database is shut down using the `NORMAL` or `IMMEDIATE` options) is called a consistent backup. In such a backup, all the database file headers are consistent with the control file, and when restored completely, the database can be opened without any recovery. When the database is operated in `NOARCHIVELOG` mode, only consistent whole database backup is valid for restore and recovery.

When the database is open and operational, the data file headers are not consistent with the control file unless the database is open in `READ ONLY` mode. When the database is shut down with the `ABORT` option this inconsistency persists. Backups of the database in such a state are termed as inconsistent backup. Inconsistent backups need a recovery to bring the database into consistent state. When databases should be available 7 days a week and 24 hours a day, you have no option but to use inconsistent backup, and this can be performed only on databases running in `ARCHIVELOG` mode.

Recovery Manager Backup Types



Backup Types Supported by Recovery Manager

There are two types of Recovery Manager backups:

- **Image copies:** Image copies are copies of a data file, or archive log file. A copy can be made using Recovery Manager or an operating system utility. The image copy of data file consists of all the blocks of the data file, including the unused blocks. The image copy can include only one file and a single operation of copy cannot be multiplexed.
- **Backup sets:** Backup sets can include one or more data files or archived logs. The output of the backup operation may comprise one or more files. You can make a backup set in two distinct ways:
 - **Full backup:** In a full backup, you back up one or more files, which is not required in an incremental backup. In a full backup, all blocks containing data for the files specified are backed up. One or more files can be included in one full backup.
 - **Incremental backup:** An incremental backup is a backup of data files that include only the blocks that have changed since the last incremental backup. Incremental backups require a base-level (or incremental level 0) backup, which back up all blocks containing data for the files specified. Incremental level 0 and full backups copy all blocks in data files, but full backups cannot be used in an incremental backup strategy.

Allocating a Channel

Allocating a Channel

- **BACKUP, RESTORE, or RECOVER command requires at least one channel.**
- **Allocating a channel starts a server process on the target database.**
- **Channels affect the degree of parallelism.**
- **Channels write to different media types.**
- **Channels can be used to impose limits.**

```

RMAN > run {
  2> allocate channel c1 type disk
  3>   format = '/u01/db01/backup/usr0520.bak';
  4> backup datafile '/u01/db01/data/user01.dbf';}

```

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

Allocating a Channel

Recovery Manager uses the channel processes to communicate between the Oracle server and the Operating system.

- An Oracle server process for the target database is created for every channel allocated. Every BACKUP, RESTORE, or RECOVER command issued in Recovery Manager requires at least one channel.
- The number of channels allocated will be the maximum degree of parallelization used during backup, restore, or recovery.
- The type of media desired determines the type of channel allocated. Query V\$BACKUP_DEVICE to determine supported device types.
- You can impose limits for the COPY and BACKUP commands by specifying parameters in the ALLOCATE CHANNEL command:
 - Read rate: Limits number of buffers read per second, per file to reduce online performance through excessive disk I/O.

```
set limit channel <name> read rate = integer
```
 - Kbytes: Limits backup piece file size created by a channel. This is useful when there are maximum file sizes for an operating system or device type.

```
set limit channel <name> kbytes= integer
```

Allocating a Channel (continued)

- **Maxopenfiles:** Limits the number of concurrently open files for a large backup (default 32). This prevents too many files being open.

```
set limit channel <name> maxopenfiles = integer
```

Allocating a Channel Examples

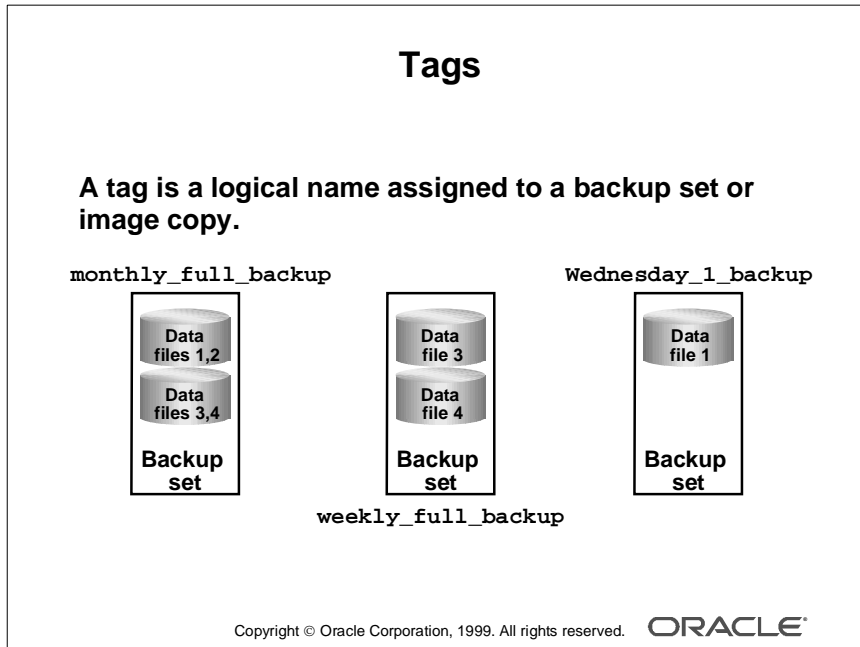
- `allocate channel for delete type disk;`

This command allocates a channel for the `CHANGE . . . DELETE` command, since a file will be removed from the disk. The `DELETE CHANNEL` command cannot be used for any other I/O operation, such as backup or copy.

- ```
RMAN> run {
 2> allocate channel d1 type disk
 3> format = '/u01/db01/backup/usr0520.bak';
 4> backup datafile '/u01/db01/data/user01.dbf';}
```

The second example allocates a channel named `d1`, where all files created by this channel will have the format `'/u01/db01/backup/usr0520.bak'`. The channel backs up one data file, `/u01/db01/data/user01.dbf`.

## Tags



## Tags

A tag is a meaningful name that you can assign to a backup set or file copy. The advantages of user tags are as follows:

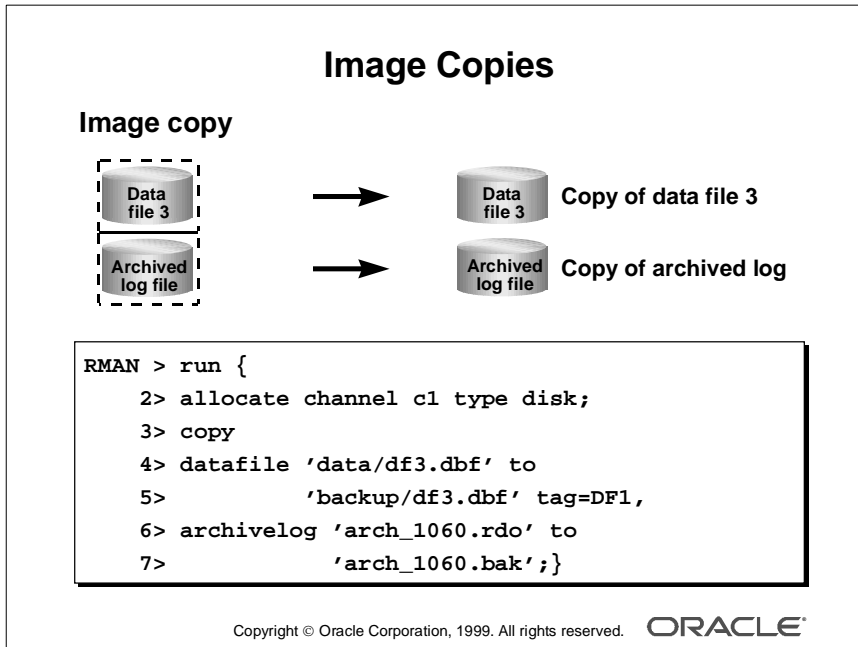
- Tags provide a useful reference to a collection of file copies or a backup set.
- Tags can be used in the LIST command to locate backed up files easily.
- Tags can be used in the RESTORE and SWITCH commands.
- The same tag can be used for multiple backup sets or file copies.

If a nonunique tag references more than one data file, then Recovery Manager chooses the most current available file.

### Example (from slide)

- Each month, a full backup of data files 1, 2, 3, and 4 is performed. The tag in the control file for this backup is `monthly_full_backup`, even though the physical filename generated is `df_DB00_863_1.dbf`.
- Each week, a full backup of data files 3 and 4 is performed. The tag name for this backup is `weekly_full_backup`.

## Image Copies



### Image Copies

You can use the Recovery Manager COPY command to create an image copy. An image produced with the Recovery Manager COPY command uses an Oracle server session to perform the task.

Recovery Manager can catalog the operating system copies of with the recovery catalog. For a data file copy made using the operating system to be useful in recovery, the corresponding tablespace should have been placed in backup mode before making the copy. This operation is important when the recovery catalog is lost and you must perform disaster recovery. Only image copies and archived logs can be cataloged.



## Image Copy Characteristics

### Characteristics of an Image Copy

- **Can be written only to a disk**
- **Can be used immediately; does not need to be restored**
- **Is a physical copy of a single data file, archived log, or control file**
- **Is most like an operating system backup (contains all blocks)**
- **Can be part of an incremental strategy**

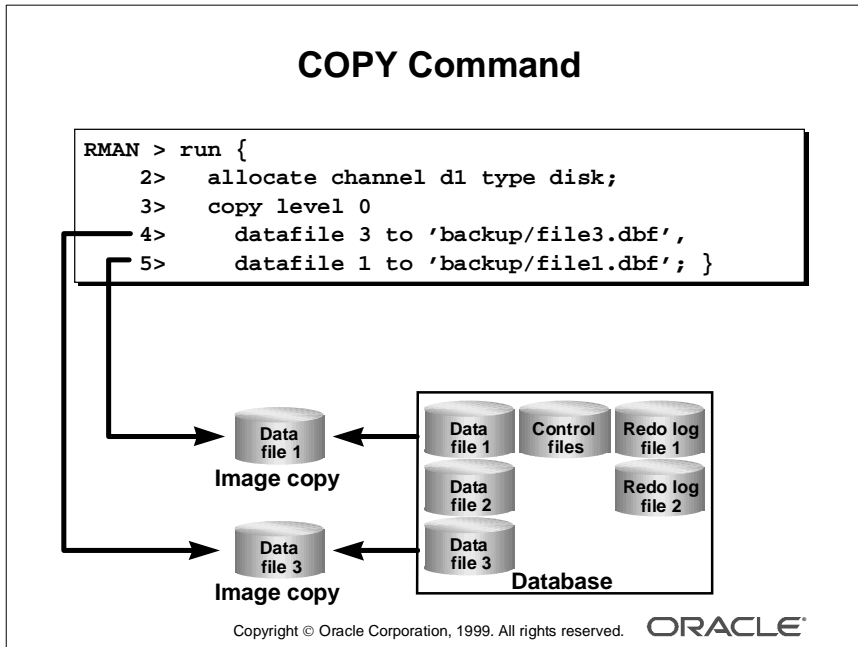
Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Characteristics of an Image Copy

An image copy has the following characteristics:

- An image copy can be written only to disk. Hence additional disk space may be required to retain the copy on the disk. When large files are being considered, copying may take a long time, but restoration time is reduced considerably because the copy is available on the disk.
- If files are stored on disk, they can be used immediately (that is, they do not need to be restored from other media). This provides a fast method for recovery using the SWITCH command in Recovery Manager, which is equivalent to the ALTER DATABASE RENAME FILE SQL statement.
- In an image copy all blocks are copied, whether they contain data or not, because an Oracle server process copies the file and performs additional actions such as checking for corrupt blocks and registering the copy in the control file. To speed up the process of copying, you can use the NOCHECKSUM parameter.
- Image copy can be part of a full or incremental level 0 backup, because a file copy always includes all blocks. Use the level 0 option if the copy will be used in conjunction with an incremental backup set.
- Image copy can be designated as a level 0 backup in incremental backup strategy, but no other levels are possible with image copy.

## COPY Command



### Copy Command

The COPY command creates an image copy of a file. The output file is always written to disk. You can copy data files, archived redo logs, or control files. In many cases, copying data files is more beneficial than backing them up, because the output is suitable for use without any additional processing.

You can execute the command from within the braces of a RUN command. Precede a COPY command with at least one ALLOCATE CHANNEL command specifying the type of disk option. You cannot make incremental copies.

During the copy operation, an Oracle server process performs checksum for each block to detect corruption. This detection is also referred to as physical corruption detection. You can use the NOCHECKSUM option to avoid the checksum operation and speed up the copy process.

You can use the CHECK LOGICAL option to test data and index blocks that pass physical corruption checks for logical corruption—for example, corruption of a row piece or index entry. If logical corruption is detected, the block is logged in the alert log and trace file of the server process.

When the number of corrupted blocks detected reaches a threshold—defined by the MAXCORRUPT clause—the copy process is terminated without populating the views.

## Image Copy Process

### Image Copy Process

- One server process operates on one file at a time.
- No blocks are skipped.
- The file is checked for corruption.
- The checksum is calculated.
- The file header is written.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE™

### Image Copy Process

When Recovery Manager copies a file, it uses the following process:

- 1 One server process operates on one file at a time.
- 2 No blocks are skipped—all blocks are included.
- 3 The file is checked for corruption.  
**Note:** V\$COPY\_CORRUPTION should be queried at the completion of every image copy.
- 4 The checksum is calculated for verifying copy integrity.
- 5 The file header is written for the new copy of the file.
- 6 An entry for the copy is recorded in the control file (and catalog if used).

## Image Copy Parallelization

### Image Copy Parallelization

#### One COPY command with many channels

```
RMAN > run {
2> allocate channel d1 type disk;
3> ...
4> allocate channel d4 type disk;
5> copy # 3 files copied in parallel
6> datafile 1 to '/disk1/df1.dbf',
7> ...
8> datafile 3 to '/disk1/df3.dbf';
9> copy # Second copy command
10> datafile 4 to '/disk1/df4.dbf'; }
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Image Copy Parallelization

By default, Recovery Manager executes each COPY command serially. However, you can parallelize the copy operation by:

- Allocating multiple channels
- Specifying one COPY command for multiple files

In the example, four channels are created, but only three will be used (channel d4 will remain idle). This is how the command is executed:

- 1 Four channels are created for writing to disk: d1, d2, d3, d4.
- 2 The first COPY command uses three channels (server processes)—one for writing each data file to disk.
- 3 The second COPY command does not execute until the previous COPY command has finished execution. It will use only one channel.

**Note:** When you use a high degree of parallelism, more machine resources are used, but the backup operation can be completed faster.

## Image Copy of All Data Files

### Image Copy of All Data Files

- **Connect to the target database using the catalog:**  

```
rman catalog rman_db1/rman_db1@rcatdb target /
```
- **Get a list of data files of the target database. If you already have the target database open, then:**  

```
RMAN> report schema;
```
- **Use the COPY command or script to copy data files:**  

```
run{
 allocate channel d1 type disk; ... # Multiple channels
 copy
 datafile 1 to '/u01/backup/df1.cpy' ...; # Multiple
 files
 release channel d1; ...
}
```
- **Use the LIST command to check if this copy is cataloged:**  

```
RMAN> list copy;
```

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### How to Make an Image Copy of All Data Files

To make an image copy of all the data files using Recovery Manager, follow this procedure:

- 1** Connect to the target database using the catalog:  

```
rman catalog rman_db1/rman_db1@rcatdb target /
```
- 2** Get a list of data files of the target database. If you already have the target database open, then:  

```
RMAN> report schema;
```
- 3** Use the COPY command or script to create the copy of all data files listed above:  

```
run{
 allocate channel d1 type disk; ...
 copy
 datafile 1 to '/u01/backup/df1.cpy' ...;
 release channel d1; ...
}
```
- 4** Use LIST command to check if this copy is cataloged:  

```
RMAN> list copy;
```

## Monitoring the Copy Process

### Monitoring the Copy Process

- **Use the COMMAND ID parameter:**

```
run {
 allocate channel t1 type disk;
 set command id to 'rman';
 copy datafile 1 to '/u01/backup/df1.cpy';
 release channel t1;}
```
- **Query V\$SESSION\_LONGOPS:**

```
SELECT sid, serial#, context, sofar, totalwork,
 round(sofar/totalwork*100,2) "% Complete"
FROM v$session_longops
WHERE opname LIKE 'RMAN:%'
AND opname NOT LIKE 'RMAN: aggregate%';
```
- **Query V\$PROCESS and V\$SESSION to get SID and SPID:**

```
SELECT sid, spid, client_info
FROM v$process p, v$session s
WHERE p.addr = s.paddr
AND client_info LIKE '%id=rman%';
```

Copyright © Oracle Corporation, 1999. All rights reserved.

ORACLE

### How to Monitor the Copy Process

To correlate a process with a channel during a backup:

- 1 Start Recovery Manager and connect to the target database and, optionally, the recovery catalog.

```
rman target / catalog rman/rman@rcat
```

- 2 Set the command id parameter after allocating the channels and then copy the desired object.

```
run {
 allocate channel t1 type disk;
 set command id to 'rman';
 copy datafile 1 to '/u01/backup/df1.cpy';
 release channel t1;}
```

- 3 Query the V\$SESSION\_LONGOPS view to get the status of the copy.

```
SELECT sid, serial#, context, sofar, totalwork
 round(sofar/totalwork*100,2) "% Complete",
FROM v$session_longops
WHERE opname LIKE 'RMAN:%'
AND opname NOT LIKE 'RMAN: aggregate%';
```

### How to Monitor the Copy Process (continued)

- 4 Using SQL\*Plus and query V\$PROCESS and V\$SESSION to get the SID and SPID. Then use an operating system utility to monitor the process or threads.

```
SELECT sid, spid, client_info
FROM v$process p, v$session s
WHERE p.addr = s.paddr
AND client_info LIKE '%id=rman%';
```

**Note:** For monitoring the copy process, you need to query the target database, and hence, the target database should be in OPEN or MOUNT state.

### Making a Data File Copy as Level 0 Incremental

This example copies the data file `tbs_01.dbf` with the `NOCHECKSUM` option to the `temp3.bak` output file, marking it as a level 0 backup:

```
run {
 allocate channel dev1 type disk;
 copy
 nochecksum
 datafile '/u01/tbs_01.dbf'
 to '/u01/backup/temp3.cpy'
 level 0;
}
```

### Copying the Control File

This example copies the current control file and gives the copy the tag `cntrl_copy`:

```
run {
 allocate channel dev1 type disk;
 copy
 current controlfile
 to '/u01/backup/cf1.cpy'
 tag = 'cntrl_copy';
}
```

## Operating System Copies

### Operating System Copies

Recovery Manager can use copies of operating system files:

```
$ svrmgrl
SVRMGR> alter tablespace system begin backup;
SVRMGR> !cp system01.dbf system01.bak
SVRMGR> alter tablespace system end backup;
SVRMGR> exit;
$ rman target sys/oracle@DB00
 rcvcat rman/rman@RCVCAT
RMAN > catalog datafilecopy 'system01.bak';
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Operating System Copies

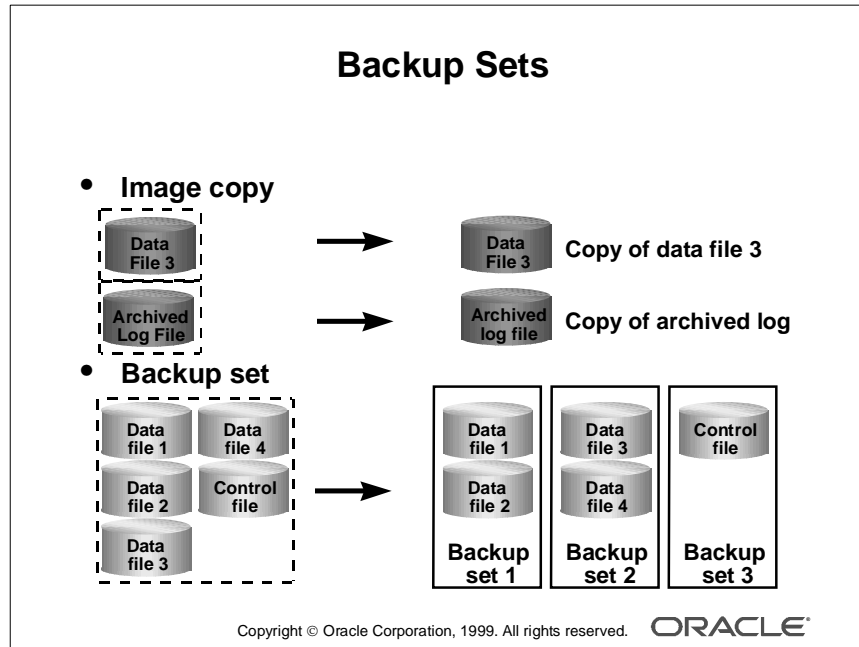
Any operating system file copy made without Recovery Manager is still a valid image copy, but it is not recognized by Recovery Manager until this copy is cataloged by Recovery Manager using a CATALOG command. Before making an online backup of a data file using the operating system, place the corresponding tablespace in backup mode for usefulness in recovery.

### Hardware Mirroring

Image copies support systems where data files are stored on mirrored disk volumes. If one disk of the mirrored disks fails, the mirror can be broken and can be used as an image copy. This reduces effort and time to restore the file. When the mirror is resilvered, remove the image copies using the CHANGE . . . DELETE command. Recovery Manager can then be notified of the new image copies using the CATALOG command.



## Backup Sets



## Backup Sets

A backup set consists of one or more physical files stored in an Oracle proprietary format, on either disk or tape. Each backup set can contain one or more Oracle files. You can make a backup set for one or more of data files, archive logs, or their copies. Backup sets can be of two types:

- **Data file:** Can contain data files and control files, but not archived logs
- **Archived log:** Contains archived logs, not data files or control files

**Note:** Backup sets may need to be restored by Recovery Manager before recovery can be performed, unlike image copies which generally are available on disks.

## Control Files in Data File Backup Sets

Each file in a backup set must have the same Oracle block size (control files and data files have the same block size, whereas archived log block sizes are machine dependent). When a control file is included, it is written in the last data file backup set. A control file can be included in a backup set either:

- Explicitly using the `include control file` syntax
- Implicitly by backing up file 1 (the system data file)

## BACKUP Command

### BACKUP Command

```
RMAN > run {
 2> allocate channel c1 type DISK;
 3> backup
 4> incremental level = 0
 5> format '/disk1/data/df_%d %s_%p.bus'
 6> (database filesperset = 2
 7> include current controlfile); }
```

**Backup set**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### BACKUP Command

The BACKUP command is executed within the braces of a RUN command. The output can be written to tape or disk. You can control the number of backup sets that Oracle produces as well as the number of input files that Recovery Manager places into a single backup set. If any I/O errors are received when reading files or writing backup pieces, the job is aborted.

When using the BACKUP command, you must do the following:

- Mount or open the target database. Recovery Manager allows you to make an inconsistent backup if the database is in ARCHIVELOG mode, but you must apply redo logs to make the backups consistent for use in recovery operations.
- Use a current control file.
- Execute the BACKUP command within the braces of a RUN command.
- Allocate a channel for execution of the BACKUP command.
- Give each backup piece a unique name. You can have a maximum of 100 backup pieces in one backup set.

You cannot combine archived redo log files and data files into a single backup. Also, when performing backups using scripts, you cannot generate unique tag names.

## BACKUP Command Options

| Option                              | Significance                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| full                                | Server session copies all blocks into the backup set, skipping only data file blocks that have never been used. The server session does not skip blocks when backing up archived redo logs or control files. Full backup is not considered in incremental backup.                                                                                                                                                                                                                                                                                                                                               |
| incremental<br>level <i>integer</i> | <p>The server session copies data blocks that have changed since the last incremental <i>n</i> backup, where <i>n</i> is any integer from 1 to 4.</p> <p>When attempting an incremental backup of level greater than 0, server process checks that a level 0 backup or level 0 copy exists for each data file in the BACKUP command.</p> <p>If you specify incremental, then in the backupSpec you must set one of the following parameters: DATA FILE, DATA FILECOPY, TABLESPACE, or DATABASE. Recovery Manager does not support incremental backups of control files, archived redo logs, or backup sets.</p> |
| filesperset<br><i>integer</i>       | <p>When you specify the filesperset parameter, Recovery Manager compares the filesperset value to a calculated value (of number of files backed up per number of channels) and takes the lowest integer of the two, thereby ensuring that all channels are used.</p> <p>If you do not specify filesperset, then Recovery Manager compares the calculated value (number of files per allocated channels) to the default value of 64 and takes the lowest of the two.</p> <p>When there are more channels than files to back up, channels remain idle. Input files cannot be split across channels.</p>           |
| skip                                | <p>Specify this parameter to exclude some data files or archived redo logs from the backup set. You have following options within the parameter.</p> <p>offline: Exclude offline data files from backup set.</p> <p>readonly: Exclude data files belonging to read-only tablespaces.</p> <p>inaccessible: Exclude data files or archived redo logs that cannot be read because of I/O errors.</p>                                                                                                                                                                                                               |
| setsize<br><i>integer</i>           | <p>Specifies a maximum size for a backup set in units of 1,024 bytes. Recovery Manager attempts to limit all backup sets to this size. Useful for backup of archive logs.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| diskratio<br><i>integer</i>         | <p>Directs Recovery Manager to assign only data files to backup sets spread across the specified number of drives. Useful for data file backups when data files are striped or reside on separate disk spindles.</p>                                                                                                                                                                                                                                                                                                                                                                                            |
| delete input                        | <p>Deletes the input files upon successful creation of the backup set. Specify this option only when backing up archived redo logs or data file copies. It is equivalent to issuing a CHANGE . . . DELETE command for all of the input files.</p>                                                                                                                                                                                                                                                                                                                                                               |

| Option                            | Significance                                                                                                                                                                                                           |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| include<br>current<br>controlfile | Creates a snapshot of the current control file and places it into each backup set produced by this clause.                                                                                                             |
| Format                            | Format of the name of output. The following format parameters can be used either individually or in combination.                                                                                                       |
| %c                                | Specifies the copy number of the backup piece within a set of duplexed backup pieces.                                                                                                                                  |
| %p                                | Specifies the backup piece number within the backup set. This value starts at 1 for each backup set and is increased by 1 as each backup piece is created.                                                             |
| %s                                | Specifies the backup set number. This number is a counter in the control file that is increased for each backup set.                                                                                                   |
| %d                                | Specifies database name.                                                                                                                                                                                               |
| %n                                | Specifies the database name, padded on the right with <i>x</i> characters to a total length of 8 characters.                                                                                                           |
| %t                                | Specifies the backup set time stamp, which is a 4-byte value derived as the number of seconds elapsed since a fixed reference time. The combination of %s and %t can be used to form a unique name for the backup set. |
| %u                                | Specifies an 8-character name constituted by compressed representations of the backup set number and the time that the backup set was created                                                                          |
| %U                                | Specifies a convenient shorthand for %u_%p_%c that guarantees uniqueness in generated backup filenames. If you do not specify a format, Recovery Manager uses %U by default.                                           |

---

## Backup Set Characteristics

### Characteristics of Backup Sets

- The **BACKUP** command creates backup sets.
- Backup sets usually contain more than one file.
- Backup sets can be written to a disk or tape.
- A restore operation is required to extract files from a backup set.
- Data file backup sets can be incremental or full.
- Backup sets do not include empty blocks.

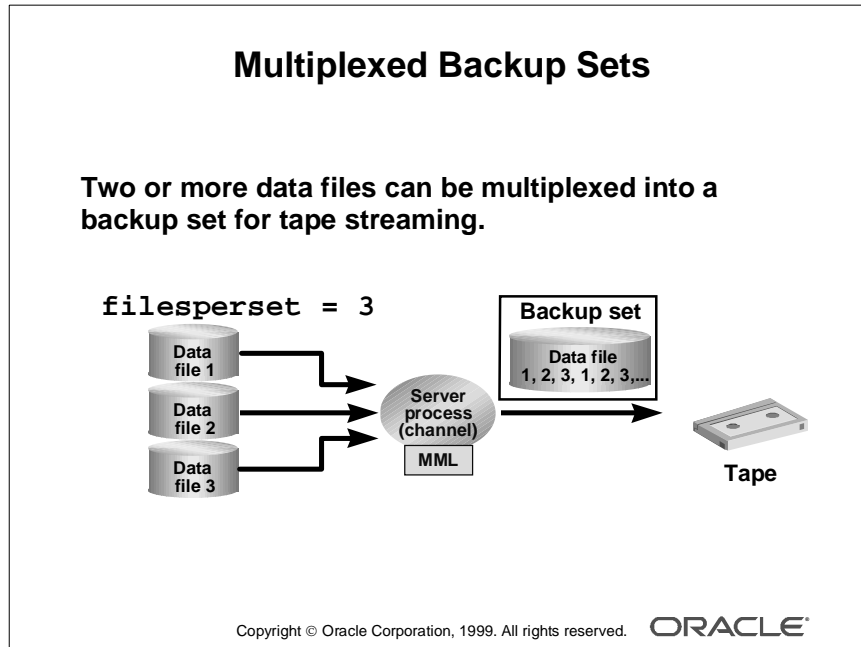
Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Characteristics of Backup Sets

A backup set is a logical structure that has the following characteristics:

- A backup set contains one or more physical files called backup pieces.
- A backup set is created by the **BACKUP** command to assist tape streaming. The **FILESERSET** parameter controls the number of data files contained in a backup set.
- For a large database, a backup set might exceed the maximum size for a single tape reel, physical disk, or operating system file. The size of each backup set piece can therefore be limited by using `set limit channel kbytes`.
- A backup set can be written to disk or tape. Oracle provides one tape output by default for most platforms, known as **SBT\_TAPE** (System Backup to Tape), which writes to a tape device when you are using a media manager. Otherwise, it simulates a tape device by writing sequentially to disk.
- A restore operation must extract files from a backup set before recovery.
- Archived log backup sets cannot be incremental (they are full by default).
- A backup set performs compression by not including empty data blocks in data files that exist beyond the high-water mark

## Multiplexed Backup Sets



### Multiplexed Backup Sets

When more than one file is written to the same backup file or piece, Recovery Manager automatically performs the allocation of files to channels, multiplexes the files, and skips any unused blocks. With a sufficient number of files to back up concurrently, high-performance sequential output devices (for example, fast tape drives) can be streamed. This is important for backups that must compete with other online system resources. It is the responsibility of the operator or storage subsystem to change the tape on the target database where the tape drive is located.

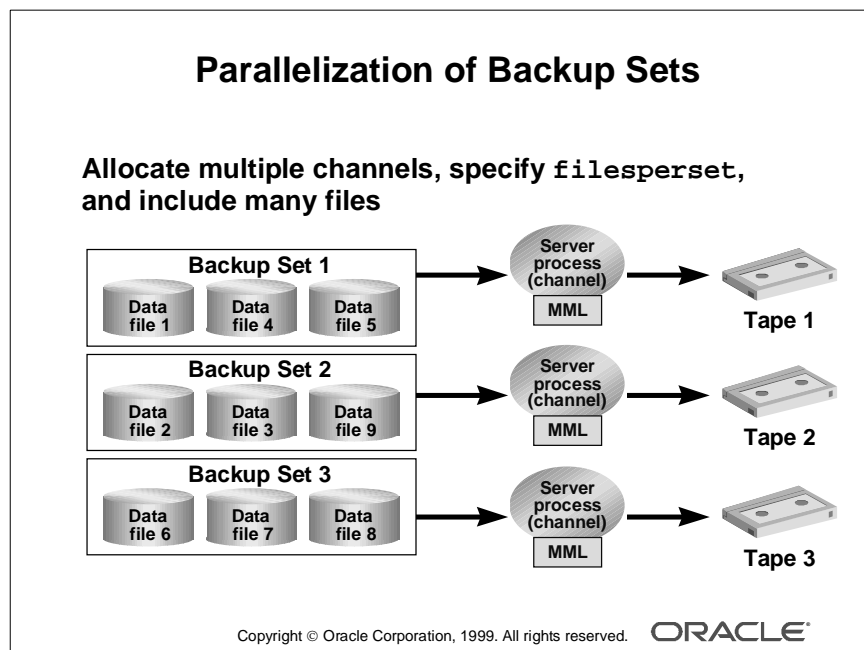
This process was designed for writing to tape but it can also be used to write to disk.

### Example

```
RMAN > run { allocate channel c1 type 'SBT_TAPE';
2> backup (database filesperset = 3); }
```

The database contains three data files that will be multiplexed together (`filesperset = 3`) into one physical file (set) and stored on tape. The data files are multiplexed by writing *n* number of blocks from data file 1, then data file 2, then data file 3, then data file 1, and so on until all files are backed up.

## Parallelization of Backup Sets



### Parallelization of Backup Sets

Parallelization of backup sets is achieved by:

- Allocating multiple channels.
- Specifying many files to back up.
- Specifying the `FILESERSET` option in the `BACKUP` command. If `FILESERSET` is not specified, only one channel is used to create one backup piece containing all files—all other channels remain idle.

### Example

- There are nine files that need to be backed up (data files 1 through 9.)
- Data files have been carefully assigned so that each set has approximately the same number of data blocks to back up (for efficiency.)
  - Data files 1, 4, and 5 are assigned to backup set 1.
  - Data files 2, 3, and 9 are assigned to backup set 2.
  - Data files 6, 7, and 8 are assigned to backup set 3.
- Because there are three files per set, there is no need to use the `FILESERSET` parameter. Three backup sets will be written each of which would contain blocks from three data files. three channels are used to write in parallel.

## Parallelization of Backup Sets (continued)

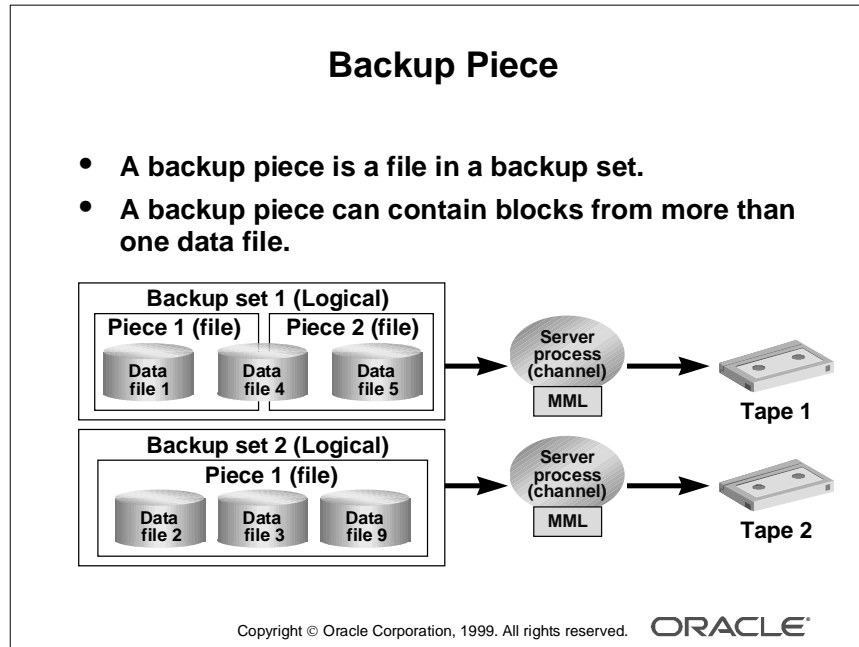
### Solution

Use the following command to achieve the specified requirements:

```
RMAN > run {
2> allocate channel c1 type disk;
3> allocate channel c2 type disk;
4> allocate channel c3 type disk;
6> backup
7> incremental level = 0
8> format '/disk1/backup/df_%d_%s_%p.bak'
9> (datafile 1,4,5 channel c1 tag=DF1)
10> (datafile 2,3,9 channel c2 tag=DF2)
11> (datafile 6,7,8 channel c3 tag=DF3);
12> sql 'alter system archive log current';
13> }
```



## Backup Piece



## Backup Piece

A logical backup set usually only has one backup piece unless specified using the SET LIMIT command. A backup piece is a single physical file that can contain one or more Oracle data files or archived logs. A backup set is a complete set of backup pieces for a full or incremental backup.

## Backup Piece Size

### Backup Piece Size

The piece size can be limited if required:

```
RMAN > run {
 2> allocate channel t1 type 'SBT_TAPE';
 3> set limit channel t1 kbytes 4194304;
 4> backup
 5> format 'df_%t_%s_%p' filesperset 3
 6> (tablespace user_data); }
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Backup Piece Size

The only way for a backup set to have more than one piece is when the following command is used:

Set limit . . . kbytes: Specifies the maximum size for a backup piece.

#### Example (from slide)

- Scenario: The USER\_DATA tablespace needs to be backed up to one tape drive. The maximum file size for the tape drive is 4 GB.
- Result: If the output file is < 4 GB, only one backup piece will be written for the backup set. If the output size is > 4 GB, more than one backup piece will be written for the backup set. Each backup piece will have blocks from three files interspersed.

## Data File Backup Process

### Data File Backup Set Process

- **Memory buffers are allocated for each file.**
- **Files are sorted in channel by descending size.**
- **Files are checkpointed and the header block is copied.**
- **Files are multiplexed together.**
- **Blocks to include are determined.**
- **Corrupt blocks are checked and checksum is calculated.**
- **Buffers are sent to the output device.**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Data File Backup Set Process

Recovery Manager performs backup of data files in the following steps:

- 1** Memory buffers are allocated for each file in the set. Each buffer is sized by `(db_block_size*db_file_direct_io_count)`.
- 2** The files to be backed up are in descending order by their size in a channel.
- 3** The files in the set are checkpointed and each file header block is copied.
- 4** Each block is checked before inclusion in the backup as follows:
  - If incremental, the SCN in block is checked to see if it qualifies for inclusion.
  - If full or level 0, the block is checked to see if it has ever contained data.
- 5** If corrupt blocks are found, this information is stored in the control file and can be queried using `V$BACKUP_CORRUPTION` after backup completion.
- 6** The checksum is calculated.
- 7** When the output buffer is filled, it is sent to the output device.

## Archived Log Backup Sets

### Archived Log Backup Sets

- Can include only archive logs
- Are always full backups

```
RMAN > run {
 2> allocate channel t1 type 'SBT_TAPE';
 3> backup filesperset 3
 4> format '/disk1/backup/ar_%t_%s_%p'
 5> (archivelog from logseq=1056 until
 6> logseq=1058 thread=1 delete input);
 7> }
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Archived Log Backup Sets

A common problem experienced by DBAs is not knowing if an archived log has been completely copied out to the archive log destination before attempting to back it up. Recovery Manager has access to control file or recovery catalog information, so it knows which logs have been archived and can be restored during recovery.

### Characteristics of Archived Log Backup Sets

- Can include only archived logs, not data files or control files.
- Are always full backups. (There is no logic in performing incremental backups, because you can specify the range of archived logs to backup.)

**Example of Archived Log Backup (from slide)** This example backs up archived logs from log sequence number 1056 to 1058 (inclusive) to a backup set, where each backup piece contains three archived logs. After the archived logs are copied, they are deleted from disk and marked as deleted in the V\$ARCHIVED\_LOG view.

## Archived Log Backup

### Archived Log Backup Process

- Archived logs are ordered by size
- Process is similar to the data file backup process
- No blocks are skipped
- Set is terminated if corruptions are found

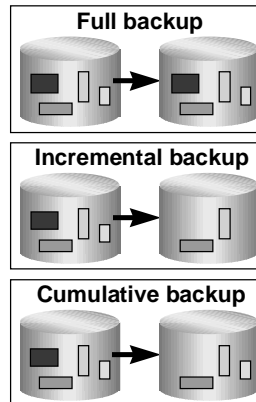
Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE<sup>®</sup>

### Archived Log Backup Process

- Archived logs in the backup set are ordered by channel and size.
- Sequence and thread numbers are replaced with a block number relative to the piece for ease of reference.
- No blocks are skipped, and there are no incremental checks.
- If any corruptions are detected, the backup process is terminated.

## Full, Incremental, and Cumulative Backups

- **Full backups** contain all data file blocks.
- **Incremental backups** contain only modified blocks from the same level.
- **Cumulative backups** contain only modified blocks from the lower or same level.



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE<sup>®</sup>

### Backup Set Scenario

- Database contains ten files.
- System contains two tape drives.
- Whole database backup is required, with three files per set required for tape streaming.
- How many backup sets are created?
- How many files are multiplexed per set?
- How many channels need to be allocated for parallel operation?

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

### Full, Incremental, and Cumulative Backups

Data file backup sets can be full, cumulative, or incremental.

**Full Backups** A backup of one or more data files that contain all blocks. A full backup can contain data files, image copies, archived logs, or control files.

**Incremental Backups (Level >= 0)** A backup of a control file or data files that contain only those blocks that have been modified since the previous incremental backup.

**Cumulative Incremental Backups (Level >= 0)** Oracle provides an option to make cumulative incremental backups. These backups reduce time by reducing the number of incremental backups needed during a restore. However, cumulative backups require more space and time because they back up information already recorded by a previous incremental backup at the same level.

## Incremental Backups

### Incremental Backups

- **Back up all changed blocks since the previous incremental**
- **Based on level 0 backup sets or image copies**
- **Write out fewer blocks than full backups**
- **Faster than performing full backups**
- **Noncumulative by default**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE<sup>®</sup>

### Incremental Backups

Incrementals have the following characteristics:

- An incremental level  $n$  backup, where  $n > 0$ , copies all changed blocks since the previous incremental (level  $\leq n$ ) backup.
- A level 0 backup set or image copy must first be created. Incremental backups are then based on changes made to these level 0 backups.
- Fewer blocks are written than when performing level 0 backups.
- Incremental backups are faster than level 0 backups.
- The default behavior of incremental backups is noncumulative.



### **Rule of Thumb**

- Perform level 0 backups if there are many updates to different blocks.
- Perform (level > 0) backups if there are many updates to fewer blocks.

There are two important SCNs:

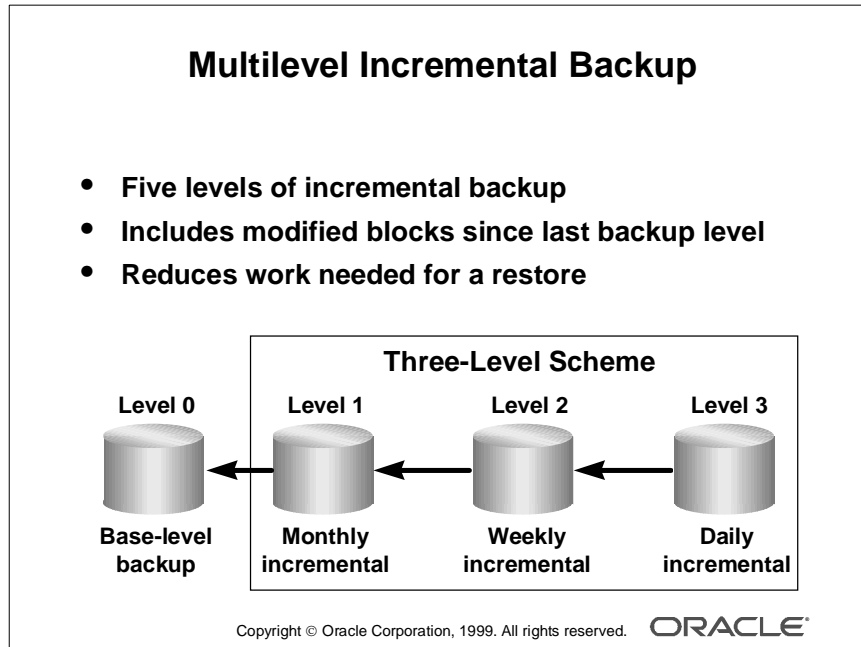
- Checkpoint\_change#:  
The data file's checkpoint SCN at the time that the backup began
- Incremental\_change#:  
Also called incremental start SCN; the checkpoint SCN for the data file as it was in the previous incremental backup (the backup this one is based on)

For an incremental level  $n$  (where  $n > 0$ ), all blocks copied include:

SCNs  $\geq$  incremental\_change#

that is, all blocks that were modified since the first file was checkpointed and blocks at the same checkpoint as the file (they may have been modified more than once at the same SCN in between being written out).

## Multilevel Incremental Backup



### Multilevel Incremental Backup

With the multilevel incremental backup feature, you can create different levels of incremental backups. You can define up to five levels, from 0 through 4.

Multilevel incremental backups facilitate recovery operations, because only one incremental backup from any particular level is needed during restore.

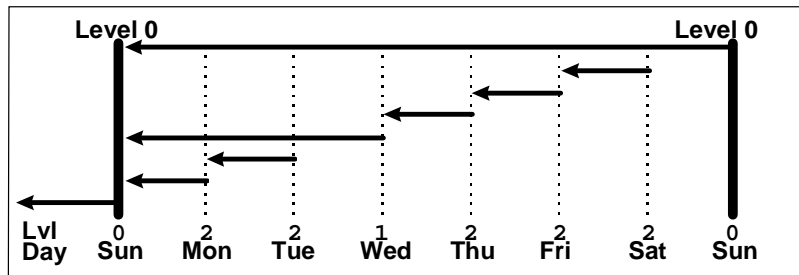
### Three-Level Backup Scheme

- A level 0 backup is performed as a basis for incremental backups.
- From this point forward, every month a level 1 incremental backup copies only data blocks changed since the last monthly backup.
- Every week, a level 2 backup copies only blocks changed since the last weekly backup.
- Every day, a level 3 backup copies only blocks changed since the last daily backup.
- If a file is lost on the fourth day, of the second week of the third month, then you need to restore only two monthly, one weekly, and three daily backups.

## Incremental Backups

### Example of an Incremental Backup

Backup that copies only Oracle blocks modified since a previous incremental



Copyright © Oracle Corporation, 1999. All rights reserved.

ORACLE

### Backup Example

You are maintaining a 100 GB database, which is continuously growing. Based on existing hardware, you determine that open backups of the entire database take 4 hours. Backups consume too much system resources during this period of time, because the database is online 24 hours a day, 7 days a week. Therefore, level 0 backups cannot be performed more than once a week, but fast recovery in case of failure is required. You therefore decide on the following backup and recovery strategy:

A level 0 backup will be performed each week on the day with the least activity. You determine this day to be Sunday.

```

RMAN> run {allocate channel c1 type disk
2> format = '/home/disk1/user4/BACKUP/sun_%s_%p.bus';
3> backup incremental level = 0 (database); }
```

Incremental level 2 backups will be performed on every other day, except Wednesday. In this way, backups will be fast, because only changed blocks from the previous day will be copied:

```

RMAN> run {allocate channel c1 type disk
2> format = '/home/disk1/user4/BACKUP/inc_%s_%p.bus';
3> backup incremental level = 2 (database); }
```

### Backup Example (continued)

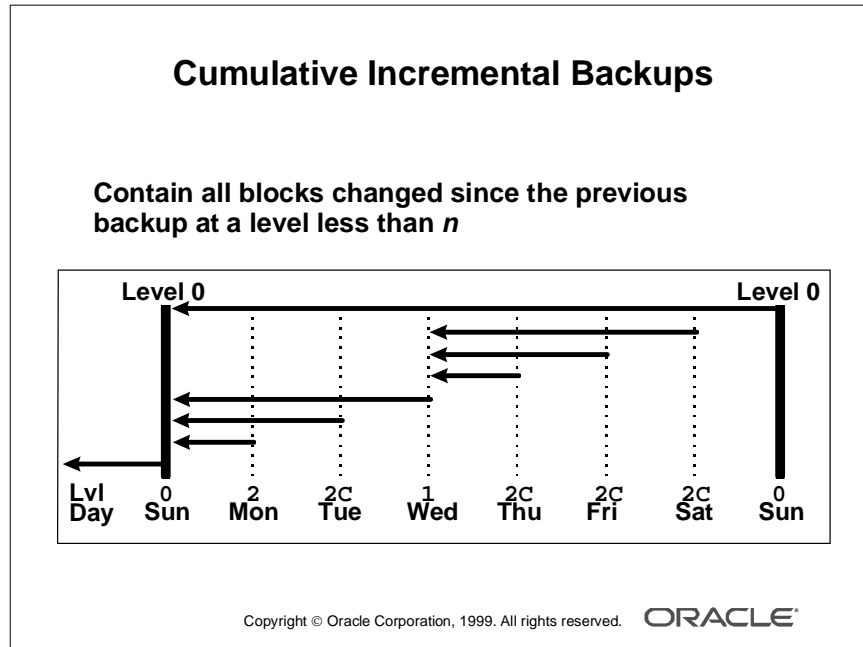
- Wednesday is a day with less database activity, so all blocks changed since Sunday are copied to assist with speed of recovery. For example, if failure occurs on Friday, then only Sunday, Wednesday, and Thursday backups need to be restored (Monday and Tuesday are not required):

```
RMAN> run {allocate channel c1 type disk
2> format = '/home/disk1/user4/BACKUP/wed_%s_%p.bus' ;
3> backup incremental level = 1 (database); }
```

- On Thursday, the incremental was replaced by a full backup. Because this does not change the backup base level, Friday's backup copy changes since Wednesday. The backup therefore can be discarded before the next level 0. If by mistake the backup on Thursday was a level 0, then the backup on Friday copies all changed blocks since Thursday, which is the new base level. This backup must now be kept until the next level 0.

**Note:** The incremental level can be set when using either the Backup Set ... or Backup Wizard ... item in Backup Manager, but not the cumulative level.

## Cumulative Incremental Backups



### Cumulative Incremental Backups

Cumulative incrementals have the following characteristics:

- A cumulative incremental level  $n$  backup (where  $n > 0$ ) copies all changed blocks since the previous incremental level  $< n$  backup.
- Cumulative incrementals back up blocks previously backed up at the same level. Therefore, they may take longer, write out more blocks, and produce larger backup files than noncumulative backups.
- Cumulative incrementals are provided for recovery speed, because fewer backups must be applied at each level when recovering.

**Example (from slide)** Cumulative backups duplicate changes already copied by the previous incremental backup at the same level. Therefore, if an incremental level 2 backup is taken, then the following cumulative level 2 backs up all newly modified blocks plus those backed up by the incremental level 2. This means that only one incremental backup of the same level is needed to completely recover.

```

RMAN> run {allocate channel c1 type disk
2> format = '/home/disk1/user4/BACKUP/cum_%s_%p.bus' ;
3> backup incremental level = 2 cumulative (database);}
```

## Backup Constraints

### Backup Constraints

- The database must be mounted or open.
- Online redo log backups are not supported.
- Only “clean” backups are usable in NOARCHIVELOG mode.
- Only “current” data file backups are usable in ARCHIVELOG mode.
- No parameter or password files are backed up.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Backup Constraints

In performing a backup using Recovery Manager, you need to be aware of the following:

- The target database instance must be available for Recovery Manager to connect. If you are not using a recovery catalog, the database must be mounted (for “cold” backups) or open (for “hot” backups).
- Backups of online redo logs are not supported. If backups of redo logs are required, the database must be run in ARCHIVELOG mode.
- If the target database is in NOARCHIVELOG mode, only “clean” tablespace and data file backups can be taken (that is, backups of “offline normal” or “read only” tablespaces). Database backups can be taken only if the database has first been shut down cleanly and restarted in mount mode.
- If the target database is in ARCHIVELOG mode, only “current” data files can be backed up (restored data files are made current by recovery).
- Recovery Manager does not back up parameter files, password files, or operating system files.
- If a recovery catalog is used, the recovery catalog instance must be open.

## Backup Set Scenarios

### Backup in NOARCHIVELOG Mode

- Ensure sufficient space for the backup.
- Shut down using the **NORMAL** or **IMMEDIATE** clause.
- Mount the database.
- Allocate multiple channels.
- Run the **BACKUP** command.
- Verify that the backup is finished and cataloged.
- Open the database for normal use.

```
RMAN> run{
2> allocate channel d1 type disk format 'C:\BACKUP\%U';
3> allocate channel d2 type disk format 'D:\BACKUP\%U';
4> backup database filesperiset 3;
5> release channel d1;
6> release channel d2;}

```

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Getting a Report of Data Files That Need Backup

The following command reports all data files that have not been backed up in the last five days.

```
report need backup days 5 database;
```

### How to Perform a Multiplexed Backup in NOARCHIVELOG Mode

- 1 Ensure that the destination directory where you want to store the backup is available and has sufficient space.
- 2 Shut down the database cleanly using the **NORMAL** or **IMMEDIATE** clause.
- 3 Mount the database.
- 4 In Recovery Manager allocate multiple channels and use a format string to multiplex channels to different disks.
- 5 Run the **BACKUP** command. Because the database is in **NOARCHIVELOG** mode, the incrementals are not applicable, so use the full backup option.
- 6 Verify that the backup is finished and cataloged.
- 7 Open the database for normal use.

## How to Perform a Multiplexed Backup in NOARCHIVELOG Mode (continued)

### Example

```
D:\>rman catalog rman4pc/rman4pc@u01.wwed125-sun target /
```

```
Recovery Manager: Release 8.1.5.0.0 - Production
```

```
RMAN-06005: connected to target database: NEW8I (DBID=3579390644)
```

```
RMAN-06008: connected to recovery catalog database
```

```
RMAN> shutdown immediate
```

```
RMAN-06405: database closed
```

```
RMAN-06404: database dismounted
```

```
RMAN-06402: Oracle instance shut down
```

```
RMAN> startup mount
```

```
RMAN-06193: connected to target database (not started)
```

```
RMAN-06196: Oracle instance started
```

```
RMAN-06199: database mounted
```

```
Total System Global Area 63803340 bytes
```

```
Fixed Size 65484 bytes
```

```
Variable Size 13332480 bytes
```

```
Database Buffers 50331648 bytes
```

```
Redo Buffers 73728 bytes
```

```
RMAN> run{
```

```
2> allocate channel d1 type disk format 'C:\BACKUP\%U';
```

```
3> allocate channel d2 type disk format 'D:\BACKUP\%U';
```

```
4> backup database filesperset 3;
```

```
5> release channel d1;
```

```
6> release channel d2;
```

```
7> }
```

```
RMAN-03022: compiling command: allocate
```

```
RMAN-03023: executing command: allocate
```



## How to Perform a Multiplexed Backup in NOARCHIVELOG Mode (continued)

### Example (continued)

```

RMAN-08030: allocated channel: d1
RMAN-08500: channel d1: sid=9 devtype=DISK

RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
RMAN-08030: allocated channel: d2
RMAN-08500: channel d2: sid=10 devtype=DISK
....

```

```
RMAN> list backup;
```

```
RMAN-03022: compiling command: list
```

#### List of Backup Sets

| Key  | Recid | Stamp     | LV | Set Stamp | Set Count | Completion Time |
|------|-------|-----------|----|-----------|-----------|-----------------|
| 2510 | 3     | 365789420 | 0  | 365789393 | 5         | 18-MAY-99       |

#### List of Backup Pieces

| Key  | Pc# | Cp# | Status    | Completion Time | Piece Name             |
|------|-----|-----|-----------|-----------------|------------------------|
| 2514 | 1   | 1   | AVAILABLE | 18-MAY-99       | C:\BACKUP\05ASR06H_1_1 |

#### List of Data Files Included

| File | Name                               | LV | Type | Ckp SCN | Ckp Time  |
|------|------------------------------------|----|------|---------|-----------|
| 2    | D:\ORACLE\ORADATA\NEW8I\RBS01.DBF  | 0  | Full | 425110  | 18-MAY-99 |
| 4    | D:\ORACLE\ORADATA\NEW8I\TEMP01.DBF | 0  | Full | 425110  | 18-MAY-99 |
| 5    | D:\ORACLE\ORADATA\NEW8I\INDX01.DBF | 0  | Full | 425110  | 18-MAY-99 |

## How to Perform a Multiplexed Backup in NOARCHIVELOG Mode (continued)

### Example (continued)

#### List of Backup Sets

| Key   | Recid | Stamp     | LV | Set Stamp | Set Count | Completion Time |
|-------|-------|-----------|----|-----------|-----------|-----------------|
| ----- | ----- | -----     | -- | -----     | -----     | -----           |
| 2511  | 4     | 365789423 | 0  | 365789394 | 6         | 18-MAY-99       |

#### List of Backup Pieces

| Key   | Pc# | Cp# | Status    | Completion Time | Piece Name             |
|-------|-----|-----|-----------|-----------------|------------------------|
| ----- | --  | --  | -----     | -----           | -----                  |
| 2515  | 1   | 1   | AVAILABLE | 18-MAY-99       | D:\BACKUP\06ASR06I_1_1 |

#### List of Data Files Included

| File  | Name                                 | LV | Type  | Ckp    | SCN       | Ckp Time |
|-------|--------------------------------------|----|-------|--------|-----------|----------|
| ----- | -----                                | -- | ----- | -----  | -----     | -----    |
| 3     | D:\ORACLE\ORADATA\NEW8I\USERS01.DBF  | 0  | Full  | 425110 | 18-MAY-99 |          |
| 6     | D:\ORACLE\ORADATA\NEW8I\OEMREP01.DBF | 0  | Full  | 425110 | 18-MAY-99 |          |
| 7     | D:\ORACLE\ORADATA\NEW8I\USER02.DBF   | 0  | Full  | 425110 | 18-MAY-99 |          |

#### List of Backup Sets

| Key   | Recid | Stamp     | LV | Set Stamp | Set Count | Completion Time |
|-------|-------|-----------|----|-----------|-----------|-----------------|
| ----- | ----- | -----     | -- | -----     | -----     | -----           |
| 2512  | 5     | 365789466 | 0  | 365789433 | 7         | 18-MAY-99       |

#### List of Backup Pieces

| Key   | Pc# | Cp# | Status    | Completion Time | Piece Name             |
|-------|-----|-----|-----------|-----------------|------------------------|
| ----- | --  | --  | -----     | -----           | -----                  |
| 2516  | 1   | 1   | AVAILABLE | 18-MAY-99       | C:\BACKUP\07ASR07P_1_1 |

## How to Perform a Multiplexed Backup in NOARCHIVELOG Mode (continued)

### Example (continued)

List of Data Files Included

| File Name                              | LV Type | Ckp SCN | Ckp Time  |
|----------------------------------------|---------|---------|-----------|
| 1 D:\ORACLE\ORADATA\NEW8I\SYSTEM01.DBF | 0 Full  | 425110  | 18-MAY-99 |

RMAN>

## How to Perform an Incremental Backup of in ARCHIVELOG Mode

Incremental backups are applicable only if the target database is open and in ARCHIVELOG mode.

- 1 Ensure that the database is in ARCHIVELOG mode by connecting as SYS in SQL\*Plus.

```
SQL> archive log list;
```

- 2 Connect to Recovery Manager and the recovery catalog.

```
rman catalog rman_db01/rman_db01@cat_db target /
```

- 3 Run the backup script or the commands. You have to have a level 0 incremental before you can use higher incrementals.

```
run{
 allocate channel d1 type disk '$HOME/INC0/%U';
 backup database incremental level 0 ;
 release channel d1;
}
```

- 4 After this at subsequent runs, you can take a higher level of incremental backup.

```
run{
 allocate channel d1 type disk '$HOME/INC1/%U';
 backup database incremental level 1 ;
 release channel d1;
}
```

## Backup of a Tablespace

```
run {
 allocate channel d1 type disk format '$HOME/INC0/TBS/%U';
 backup incremental level 0 tablespace SYSTEM;
 release channel d1;
}
```

## Backup of Archived Log Files

The following command backs up to tape all the archive logs created during the last 24 hours and removes them from the disk.

```
run {
 allocate channel dev1 type disk '$HOME/INCL/%U';
 backup archivelog from time 'SYSDATE-1' delete input;
 release channel dev1;
}
```

Use the following command to list all files in the backup set:

```
RMAN> list backupset of database tag=SCEN;
RMAN-03022: compiling command: list
RMAN-06230: List of Datafile Backups
RMAN-06231: Key File Type LV Completion_time ...
RMAN-06232: ---- ---- ----- -- -----
RMAN-06233: 1744 1 Incremental 0 25-APR-99 ...
RMAN-06233: 1744 2 Incremental 0 25-APR-99 ...
RMAN-06233: 1744 3 Incremental 0 25-APR-99 ...
...
```

## Backups Using Stored Scripts

### Backups Using Stored Scripts

Execute the CREATE SCRIPT command:

```
create script BackupExample {
 allocate channel t1 type 'SBT_TAPE';
 allocate channel t2 type 'SBT_TAPE';
 backup
 filesperset 3
 format '/disk1/backup/df_%d_%s_%p'
 tag=SCEN
 (database include current controlfile); }
```

Run the script:

```
run { execute script BackupExample; }
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Backups Using Stored Scripts

An alternative to placing the entire syntax into a RUN command is creating a script file using the CREATE SCRIPT syntax. A script is stored in the recovery catalog using this name after Recovery Manager has executed the command.

## Backups Using Stored Scripts (continued)

Using the previous example:

- 1 Create a script named FULLBACKUP that will be used to perform full backup of the database. You should first create a text file that can be executed to create the script. The text file then can be edited to create further scripts as necessary with minor changes to the text file.

```
create script fullbackup{
allocate channel t1 type disk format '$HOME/INC0/%U';
allocate channel t2 type disk format '$HOME/INC1/%U';
backup
 filesperset 5 tag=SCEN
 (database include current controlfile); }.
```

- 2 Execute the script using the CMDFILE option when connecting to Recovery Manager.

```
rman catalog rman_db01/rman_db01@catdb target / cmdfile
fullback.rmn log scrpt.log
exit;
```

- 3 This script is now created in the recovery catalog as FULLBACKUP.

- 4 Each time you need to do a full backup, issue the following run command:

```
RMAN > run { execute script FULLBACKUP; }
RMAN-03021: executing script: BackupExample
RMAN-03022: compiling command: allocate
RMAN-03023: executing command: allocate
...Use the following command to check for block level corruption:
SQL> select * from v$backup_corruption where file# in (1,2,...);
no rows selected
```

## Result

- Four sets are created, each with three multiplexed files.
- These sets are created in parallel, if two channels are allocated. If the keyword `filesperset` is not included, only one channel is used, and that channel will create one piece that contains all ten files. The other channels will be idle.

## Miscellaneous Issues

### Miscellaneous Issues

- **Terminating a Recovery Manager job**
- **Backing up the control file frequently**
- **Recording corrupt data file blocks in the control file and in the alert log**
- **Changing a fractured block while Recovery Manager is writing it**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE™

### Terminating a Recovery Manager Job

Recovery Manager records only backup sets in the control file that have finished successfully. If a Recovery Manager job terminates abnormally, incomplete files might exist at the operating system. Recovery Manager will not use them, but you will need to remove them.

### Backing Up the Control File Frequently

Always back up the control file with any data file backup. If a control file needs to be restored, and incremental backups have been taken, archived logs need to be restored to recover the control file. The benefit of taking incremental backups for speed of recovery, rather than applying large numbers of archived logs, is therefore lost.

**Note:** Creating a backup set with data file number 1 includes the control file.

## Detecting Corruption

Recovery Manager detects and can prohibit any attempt to perform operations that would result in unusable backup files or corrupt restored data files.

Information about corrupt data file blocks encountered during a backup are recorded in the control file and the alert log. The server identifies corrupt data file blocks, but they are still included in the backup. The Oracle server records the address of the corrupt block and the type of corruption in the control file. To view corrupt blocks from the control file, view either V\$BACKUP\_CORRUPTION for backup sets or V\$COPY\_CORRUPTION for image copies.

To limit the number of previously undetected block corruptions allowed (2 in this example) for a data file backup, use the `set maxcorrupt` syntax:

```
RMAN > run {
 2> allocate channel c1 type disk;
 3> set maxcorrupt for datafile '/disk1/data/oem_01.dbf' to 2;
 4> copy level 0 datafile '/disk1/data/oem_01.dbf'
 5> to '/disk1/backup/oem_01.dbf';
 6> release channel c1; }
```

## Changing a Fractured Block

A fractured block is a block that was read by the backup utility while it was being written, and thus is inconsistent. Checksums allow the backup utility to detect fractured blocks and reread them to obtain a consistent version. This is the default operation.



## Memory Usage by Recovery Manager

### Memory Usage by Recovery Manager

- Input and output buffers are allocated by Recovery Manager.
- I/O slaves can be used if the operating system supports asynchronous I/O or not.
- When I/O slaves are used, buffers are allocated out of the large pool.
- Query V\$SGASTAT for memory usage.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Memory Usage by Recovery Manager

When copy or backup command is executed, memory buffers are needed. You can configure the memory based on the following assumptions:

- Each data file has four input buffers and each channel has four output buffers (sized by `db_block_size*db_file_direct_io_count`).
- Memory will be allocated from the PGA if I/O slaves are not used. Otherwise memory is obtained from the large pool. Configure the large pool using the `LARGE_POOL_SIZE` parameter.
- When the operating system supports asynchronous I/O (AIO), set up I/O slaves using the following parameters:
  - `DISK_ASYNCH_IO = TRUE`
  - `TAPE_ASYNCH_IO = TRUE`
  - `BACKUP_TAPE_IO_SLAVES = TRUE`
- To set up a dedicated slave to write to a tape device, set the following parameter. The slave process will simulate an asynchronous I/O on a platform that does not support asynchronous I/O :
  - `BACKUP_TAPE_IO_SLAVES = TRUE`
- Setting I/O slaves requires an additional process, so allocate resources accordingly.

### Viewing Memory Usage

An appropriate number of disk I/O slaves is 2 or 3. If more are needed, it is probably be more effective to allocate another channel. Memory usage can be viewed using `V$SGASTAT` where `name = 'KSFQ buffers' ;`

If there is not enough space in the large pool, the I/O slaves that have already started up will continue, and the remaining processes will run synchronously. The processes that were unable to get space in the large pool will report this in the target database's alert log.

## Troubleshooting

### Troubleshooting Recovery Manager

- Use V\$SESSION\_LONGOPS to check backup progress.
- Check V\$SESSION\_WAIT to see wait events.
- Investigate other problems:
  - Failed to start RPC call
  - Backup hanging at beginning
  - No tape device found

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Troubleshooting

- Use V\$SESSION\_LONGOPS to check the progress of a backup. For example:

```
SQL> select sid, serial#, context,
2 round(sofar/totalwork*100,2) "% Complete",
3 substr(to_char(sysdate,'ymmdd hh24:mi:ss'),1,15) "Time"
4 from v$session_longops
5 where compnam = 'dbms_backup_restore';
```

| SID  | SERIAL# | CONTEXT | % Complete | Time            |
|------|---------|---------|------------|-----------------|
| ---- | -----   | -----   | -----      | -----           |
| 19   | 86      | 46      | 100        | 971124 05:13:40 |
| 19   | 86      | 1060    | 100        | 971124 05:13:40 |
| 29   | 181     | 64      | 10.56      | 971124 05:13:40 |

If the % Complete is not constantly increasing, there may be a problem.

### Troubleshooting (continued)

- Check V\$SESSION\_WAIT to see what events are being waited for:

```
SQL> select event, pltext, seconds_in_wait
 2 from v$session_wait
 3 where wait_time = 0;
```

**Backup Hanging at Beginning** After the tape backup has begun, if Recovery Manager has not output new information, V\$SESSION\_WAIT will not show any information for the COMPNAM = 'DBMS\_BACKUP\_RESTORE' statement, and the backup may seem hung. In this situation, check the media manager to ensure that it has not hung or terminated abnormally.

Check the sbtio.log file for more information, because most media managers write information to this file. It is located in \$ORACLE\_HOME/rdbms/log directory (not background\_dump\_dest or user\_dump\_dest).

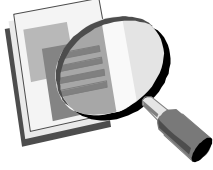
**No Tape Device Found** If the following errors are detected, the 7004 error usually indicates that no tape device is found.

```
RMAN-03007: exception occurred ... error is retryable
RMAN-07004: unhandled exception ... on channel c4
RMAN-10032: unhandled exception ... step 4: ORA-06512:at line 158
RMAN-10035: exception raised ... operation failed, retry possible
ORA-19506: failed to create sequential file, name="df_98_1",
parms=""
ORA-27007: failed to open file
...
```

## Data Dictionary Views

### Data Dictionary Views

- **V\$ARCHIVED\_LOG**
- **V\$BACKUP\_CORRUPTION**
- **V\$COPY\_CORRUPTION**
- **V\$BACKUP\_DATAFILE**
- **V\$BACKUP\_REDOLOG**
- **V\$BACKUP\_SET**
- **V\$BACKUP\_PIECE**



Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Data Dictionary Views

Recovery Manager data dictionary views used to query the control file are:

- **V\$ARCHIVED\_LOG**: Shows which archives have been created, backed up, and cleared in the database.
- **V\$BACKUP\_CORRUPTION**: Shows which blocks have been found corrupt during a backup of a backup set.
- **V\$COPY\_CORRUPTION**: Shows which blocks have been found corrupt during an image copy.
- **V\$BACKUP\_DATAFILE**: Useful for creating equal sized backup sets by determining the number of blocks in each data file. Can also find the number of corrupt blocks for the data file.
- **V\$BACKUP\_REDOLOG**: Shows archived logs stored in backup sets.
- **V\$BACKUP\_SET**: Shows backup sets that have been created.
- **V\$BACKUP\_PIECE**: Shows backup pieces created for backup sets.

## Summary

### Summary

In this lesson, you should have learned that:

- An image copy is a single file that can be copied only to a disk.
- Backup sets multiplex several data files.
- Backup sets usually contain only one backup piece (unless `SET LIMIT` is used).
- Incremental and cumulative backups can decrease backup and recovery time.
- `SBT_TAPE` writes to tape when using a media manager.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

## Quick Reference

| Context                   | Reference                                                                                                                                                                                                               |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameters                | BACKUP_TAPE_IO_SLAVES<br>DB_BLOCK_SIZE<br>DB_FILE_DIRECT_IO_COUNT<br>DISK_ASYNC_IO<br>LARGE_POOL_SIZE<br>SESSIONS<br>TAPE_ASYNC_IO                                                                                      |
| Dynamic performance views | V\$ARCHIVED_LOG<br>V\$BACKUP_CORRUPTION<br>V\$COPY_CORRUPTION<br>V\$BACKUP_DATAFILE<br>V\$BACKUP_REDOLOG<br>V\$BACKUP_SET<br>V\$BACKUP_PIECE<br>V\$BACKUP_DEVICE<br>V\$SESSION_LONGOPS<br>V\$SESSION_WAIT<br>V\$SGASTAT |
| Data dictionary views     | None                                                                                                                                                                                                                    |
| Commands                  | RUN { commands; }<br>ALLOCATE CHANNEL ...;<br>COPY ...;<br>BACKUP ...<br>CREATE SCRIPT { commands; }<br>RUN { EXECUTE SCRIPT name; }                                                                                    |





## **Restoration and Recovery Using Recovery Manager**

## Objectives

### Objectives

**After completing this lesson, you should be able to do the following:**

- **Restore and recovery considerations using RMAN**
- **Restore database in NOARCHIVELOG mode**
- **Restore and recover a tablespace**
- **Restore and recover a datafile**
- **Incomplete recovery using RMAN**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

## Restoration and Recovery Using Recovery Manager

### Restoration and Recovery Using RMAN

- **Consider the available backups and their location.**
  - If image copies are available on the disk, you can avoid having to restore files.
  - For large files, this saves time at the cost of disk space.
- **If incremental backups are on tape, restore required files using RMAN and perform recovery. RMAN applies the latest level 0 backup and then incrementals. Finally it applies the archived logs.**
- **You can perform recovery at the following levels: database, tablespace, and datafile.**
- **When using recovery catalog, you can:**
  - **Perform tablespace point-in-time recovery**
  - **Recover to a previous incarnation**

Copyright © Oracle Corporation, 1999. All rights reserved.

**ORACLE**

### Restoring and Recovering Data

When performing restore and recovery operations, you have to consider the available backups and their location.

If image copies are available on the disk, you can avoid having to restore files you could switch to the copy file with and apply recovery. When large data files are considered this can be a very effective saving. But this entails a cost in terms of disk space.

If you used an incremental strategy for backup, you may have to restore required files using RMAN and perform recovery. While performing recovery, RMAN automatically chooses the latest level 0 backup and applies the incrementals from there of. Finally, it applies the archived logs.

Once again, if backups are available on the disk, RMAN uses server processes to apply restored backups and archive logs from the disk based on the information in the catalog or control file.

Using RMAN you can perform recovery at the following levels:

- Database
- Tablespace
- Data file

You can also perform tablespace point-in-time recovery using RMAN. When you use the recovery catalog, you can recover the database to a previous incarnation.

## Restoring a Database in NOARCHIVELOG Mode

### Restoring a Database in NOARCHIVELOG Mode

```
rman target / nocatalog
RMAN> startup mount
RMAN> run {
2> allocate channel d1 type disk;
3> allocate channel d2 type disk; # Parallel Restore
4> restore database;
5> recover database;
6> sql "alter database open";}
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Restoring a Database in NOARCHIVELOG Mode

When restoring, RMAN uses the recovery catalog or target database control file to decide which full or incremental backups, archived logs, or image copies it will use. Remember, for a database in noarchivelog mode, all database files need to be restored.

#### Example

This example assumes that:

- A full backup taken using RMAN is available on disk.
- The control files are not damaged and hence not restored.

```
rman target / nocatalog
RMAN> startup mount
RMAN> run {
2> allocate channel d1 type disk;
3> allocate channel d2 type disk; # For Parallel Restore
4> restore database;
5> recover database;
6> sql "alter database open";}
```

### **Considerations for Restoring in NOARCHIVELOG Mode**

Following are some considerations when restore or recovery operations are performed on a database set in NOARCHIVELOG mode:

- To restore to a previous point in time, you may have to use the backup of older control file and use the **RESTORE CONTROL FILE** option. The database should be in **NOMOUNT** state to restore the control file.
- The target database must be in mount mode for restoration of data files.
- All the data files should be restored from a backup taken at the same time.
- The **ALTER DATABASE OPEN RESETLOGS** command may be required if a backup of the control file was restored.
- A full backup is required after an **OPEN WITH RESETLOGS** command.
- You can only restore using **RMAN** if the backups were taken or registered with **RMAN**. The recovery catalog is not mandatory, but is advisable.

## Restoring Data Files to a Different Location

### Restoring Data Files to a Different Location

- Media failure may necessitate a relocation of files.
- Connect to RMAN.

```
rman target / catalog rman_db1/rman_db1@catdb
```

- Mount the database.

```
RMAN > startup mount
```

- Restore the datafiles to the new location. Record the change in the control file.

```
run{
 allocate channel d1 type disk;
 set newname for datafile 1 to
 '<newdir>/system01.dbf' ...
 restore database;
 switch datafile all;
 recover database;
 sql "alter database open"; }
```

Copyright ©Oracle Corporation, 1999. All rights reserved.

ORACLE

### Restoring Data Files to a Different Location

A relocation of data files may be needed while restore operations are being performed, especially if recovery is necessitated by media failure.

- 1 Connect to RMAN.

```
rman target / catalog rman_db1/rman_db1@catdb
```

- 2 Mount the database.

```
RMAN > startup mount
```

- 3 Restore the data files to the new location. Record the change in the control file.

```
run{
 allocate channel d1 type disk;
 set newname for datafile 1 to '<newdir>/system01.dbf' ...
 restore database;
 switch datafile all;
 recover database;
 sql "alter database open"; }
```

## Recovering a Tablespace

### Recovering a Tablespace

- Check the status of the database.
- List data files that need to be restored.
- Run the RMAN commands to restore and recover.
- Check the completion of recovery.
- Take a backup.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE<sup>®</sup>

### How to Recover a Tablespace

#### 1 Check the status of database.

Query the V\$INSTANCE view to note the status of database.

```
select instance_name, status, archiver from v$instance;
```

#### 2 List the data files that need to be restored and recovered.

Query the V\$DATAFILE\_HEADER.

```
select name, file#, tablespace_name, status, error, recover,
 fuzzy
from v$datafile_header
where tablespace_name = 'TBS1';
```

If the error column is not null, it indicates that there may be a need to recover the tablespace by restoring the data files or switching data files to the image copy of that file.

## How to Recover a Tablespace (continued)

**3** Run the RMAN commands.

```
run{
 allocate channel dev1 type disk;
 sql "alter tablespace tbs1 offline immediate";
 restore tablespace tbs1;
 recover tablespace tbs1;
 sql "alter tablespace tbs1 online";
 release channel dev1;
}
```

**4** Check to see that the tablespace has been recovered.

**5** Take an appropriate backup.

**Note:** The database should be in ARCHIVELOG mode for open database recovery. While it is not mandatory, it is preferable to enable automatic archiving.



## Relocating a Tablespace

### Relocating a Tablespace

- Check the location and size of the data file.
- Make sure the tablespace is offline.
- Use **SET NEWNAME** for restore and **SWITCH** to record in control file. Then use the **RECOVER** command.
- When recovery is finished, bring the tablespace online.

```
run{
 allocate channel dev1 type disk;
 sql "alter tablespace tbs1 offline immediate";
 set newname for datafile 2 to 'disk1/data/df2.dbf';
 restore (tablespace tbs1);
 switch datafile 2; # Update control file, catalog
 recover tablespace tbs1; # Recover the tablespace
 sql "alter tablespace tbs1 online";
 release channel dev1;}

```

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Relocating a Tablespace

If a data file cannot be accessed because of disk failure, it probably must be restored to a new location or switched to an existing image copy.

The same procedure is useful when you want to relocate the tablespace, because you are running short of disk space in one drive or you are reorganizing the database to improve performance.

You notice that disk 2 has been corrupted and the database is still open. Occasionally users complain that they cannot access information in data file number 2:

### How to Restore Data Files to a New Location

Following is the procedure to restore the data files to a new location:

- 1 Check the location and size of the data file on disk 2, using the following command

```
SQL> select file#, name, bytes from dba_data_files;
```

| FILE# | NAME                      | BYTES    |
|-------|---------------------------|----------|
| 1     | /disk1/data/system_01.dbf | 31457280 |
| 2     | /disk2/data/df2.dbf       | 10485760 |
| ...   |                           |          |

You determine that there is enough space on disk 1 for data file 2.

### How to Restore Data Files to a New Location (continued)

- 2 Make sure the file (tablespace if necessary) is offline so that it can be restored successfully using the RESTORE command.
- 3 Since the file was copied to a new location (using SET NEWNAME), the file must be made current by notifying the Oracle server of the new file location using the SWITCH command.
- 4 Use the RECOVER command to start applying the combination of archives, incrementals, cumulatives, and redo logs to the restored file to synchronize the database.
- 5 When recovery is finished, bring the data file online.

Notify users that the database is available for use, and that they should reenter any data that was not committed before system failure. The following command may be used:

```
run{
 allocate channel dev1 type disk:
 sql "alter tablespace tbs1 offline immediate";
 set newname for datafile '/disk2/data/df2.dbf'
 to 'disk1/data/df2.dbf'; # Specify where to restore the file
 restore (tablespace tbs1);# Restore the datafile from tape
 switch datafile 2; # Update the control file and recovery catalog
 recover tablespace tbs1; # Recover the tablespace
 sql "alter tablespace tbs1 online";
 release channel dev1;}
```

When restoring, RMAN uses the recovery catalog or target database control file to decide which full or incremental backups, archived logs, or image copies it will use. Remember, for a database in ARCHIVELOG mode, only lost files need to be restored.

### SET NEWNAME Command

When restoring many data files to a new location, use the SET NEWNAME command:

```
set newname for datafile <name> to <newname>;
```

The SWITCH command is used to make the restored files current.

```
switch datafile <name>;
```

### **Using Recovery Manager to Recover**

Once a file is restored, it needs to be recovered. RMAN has more choices than just restoring a data file and applying archives.

When restoring files for a database in ARCHIVELOG mode, you need to be aware of the following situations:

- The files being restored must be offline.
- RMAN can restore only the backups that were registered with RMAN.

## Incomplete Recovery of a Database

### Incomplete Recovery of a Database Using RMAN

- Mount the database.
- Allocate multiple channels to assist parallelization.
- Restore all data files.
- Recover the database using UNTIL TIME, CANCEL, or CHANGE.
- Open the database using RESETLOGS.
- Perform a whole database backup.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Incomplete Recovery of a Database Using RMAN

The restore and recovery process for incomplete recovery follows the same procedure and syntax as complete recovery, except that all data files need to be restored from the past backup.

#### Recover Syntax

A RECOVER command is usually issued within a RUN or CREATE SCRIPT command:

```
recover <Object>;
where: Object database <until_clause>
 tablespace <until clause>
```

#### Note

- This is not the full syntax.
- The target database must be in mounted state.
- The files being restored must be offline and you can only restore using RMAN if the backups were taken or registered with RMAN.

## Incomplete Recovery of a Database: Example

### Incomplete Recovery Using RMAN

```
RMAN > run {
 > allocate channel c1 type DISK;
 > allocate channel c2 type DISK;
 > set until time = '1997-12-09:11:44:00';
 > restore database;
 > recover database;
 > sql "alter database open resetlogs"; }
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### How to Perform an Incomplete Recovery

At 12:00 p.m. on Tuesday, December 9, 1997, you immediately shut down the database and begin recovery after determining that the employee table was dropped. The approximate time of failure is known and the database structure has not changed since 11:44 a.m. You can use the UNTIL TIME method:

- 1 If the target database is open, perform a clean shut down.
- 2 Mount the target database. Do not back up the database during the recovery.
- 3 Start Recovery Manager and connect to the target database, preferably using a recovery catalog. Before starting RMAN, ensure that NLS\_LANG and NLS\_DATE\_FORMAT environment variables are set appropriately:

```
$ NLS_LANG=american
```

```
$ NLS_DATE_FORMAT='YYYY-MM-DD:HH24:MI:SS'
```

```
$ rman target rman/rman@DB00 rcvcat rman/rman@RCVCAT
```

- 4 Create the RUN command which will perform the restore and recovery:

```
RMAN> run { allocate channel c1 type DISK;
```

- 5 Restore all data files from a backup which will enable successful incomplete recovery. RMAN will choose these files based on the SET UNTIL command:

```
RMAN> ... set until time = '1997-12-09:11:44:00';
```

```
RMAN> ... restore database;
```

**Note:** The RESTORE DATABASE database command only restores datafiles, not control files, redo logs, or archived logs.

### How to Perform an Incomplete Recovery (continued)

- 6** You may need to restore archived logs. If there is enough space available, restore to the LOG\_ARCHIVE\_DEST location or use the SQL ALTER SYSTEM ARCHIVE LOG START TO <location> or RMAN SET ARCHIVELOG DESTINATION TO <location> commands to change the location.
- 7** Recover the database to the time specified in the SET UNTIL command:  

```
RMAN> ... recover database;
```
- 8** Open the database using the RESETLOGS option:  

```
RMAN> ... sql "alter database open resetlogs"; }
```
- 9** Check that the table exists and perform a backup.
- 10** Notify users that the database is available for use, and that they should reenter any data that was not committed before system failure.
- 11** If using a recovery catalog, register the new incarnation of the database:  

```
RMAN> reset database;
```

## Restoring a Database to a Previous Incarnation

### Restoring a Database to a Previous Incarnation

- List the incarnation of the database.
- Shut down the database and move current files to another location.
- Start up the database in the NOMOUNT state.
- Reset the database to a previous incarnation.
- Restore the control file from the old incarnation.
- Mount the database.
- Restore data files.
- Recover the database.
- Open the database with RESETLOGS.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### How to Restore a Database to a Previous Incarnation

You may be required to reset the database to a previous incarnation when some data from the previous incarnation is essential.

- 1 List the incarnation of database. Notice the current incarnation.

```
RMAN> list incarnation of database;
```

List of Database Incarnations

| DB Key | IncKey | DB Name | DB ID      | CUR Reset | SCN    | ResetTime |
|--------|--------|---------|------------|-----------|--------|-----------|
| -----  | -----  | -----   | -----      | -----     | -----  | -----     |
| 1      | 2      | U37     | 2220934682 | NO        | 33884  | 20-MAY-99 |
| 1      | 165    | U37     | 2220934682 | YES       | 158443 | 25-MAY-99 |

- 2 Shut down the database and move the current data files and control file to another location.

```
RMAN> shutdown immediate;
```

```
RMAN> host;
```

```
$> mv <datafile> BACKUP
```

```
exit
```

```
RMAN>
```

- 3 Start the instance in the NOMOUNT state.

```
RMAN> startup nomount;
```

## How to Restore a Database to a Previous Incarnation (continued)

- 4** Reset the database to the previous incarnation.

```
RMAN> reset database to incarnation 2;
```

- 5** Restore the control file from the previous incarnation.

```
RMAN> run {
 2> allocate channel d1 type disk;
 3> restore controlfile; }
RMAN>
```

- 6** Mount the database.

```
RMAN> alter database mount;
```

- 7** Restore the database. The control file now in use is from the previous incarnation and will contain information about backups of the previous incarnation. Ensure that the backup of data files from the previous incarnation is available appropriately.

```
RMAN> run {
 allocate channel d1 type disk;
 restore database;}
RMAN>
```

- 8** Recover the database. Then open the database with RESTLOGS.

```
RMAN> run{
 allocate channel d1 type disk;
 recover database;
 sql "alter database open resetlogs"; }
RMAN>
```

- 9** Check whether the necessary data is available and inform the users.



## Summary

### Summary

**In this lesson, you should have learned how to do the following:**

- **Use RMAN to restore and recover a database in NOARCHIVELOG mode**
- **Recover a data file in an open database**
- **Recover a tablespace—disk failure**
- **Perform an incomplete recovery**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**



## **Oracle Standby Database**

## Objectives

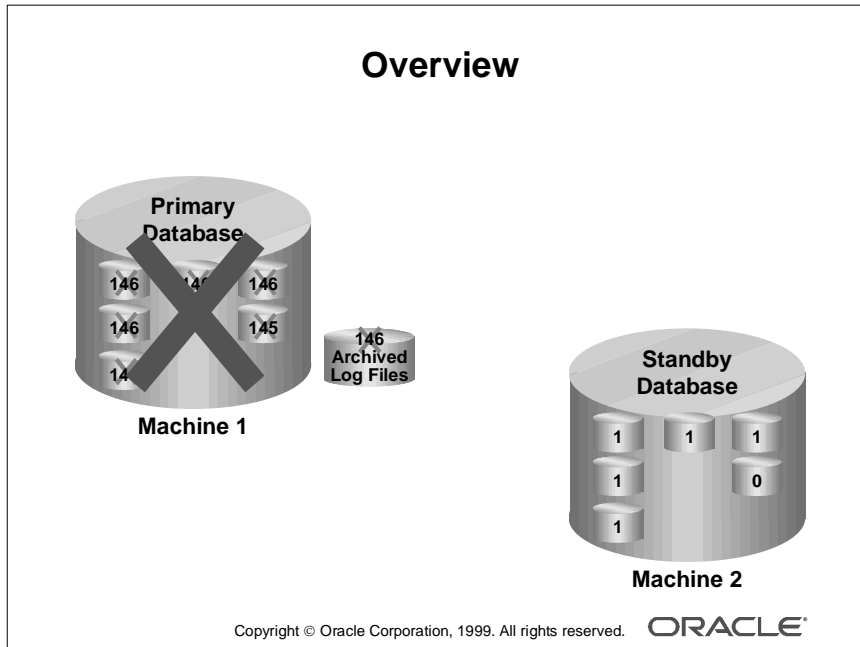
### Objectives

**After completing this lesson, you should be able to do the following:**

- **Explain the use of a standby database**
- **Configure initialization parameters**
- **Create, maintain, and activate a standby database**
- **Describe Managed Recovery mode**
- **Set a standby database in Read Only mode**
- **Describe the process of propagating structural changes to a standby database**
- **Describe the impact of nologging actions on the primary database**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

## Overview



## Overview

Database administrators can maintain a duplicate, or standby, copy of a database at a remote site to provide continued primary database availability in the event of a failure. The standby database is created with a special copy of the control file from the primary database. The standby database is kept in close synchronization with the primary database by applying the primary database's archived log files to the standby database. It is therefore necessary to operate the primary database in ArchiveLog mode to avail the benefit of a standby database.

## Standby Database Features

### Features of a Standby Database

**A standby database:**

- **Is a copy of your primary database on a separate machine**
- **Is kept in recovery mode for immediate activation**
- **Can be in Managed Recovery mode**
- **Can be opened in Read Only mode for queries**
- **Becomes the primary database after activation**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE™

### Standby Database Features

A standby database is generally used when high availability of data is required. A standby database is one way of quickly providing access to data if the primary database fails and recovery will take longer than the desired time.

A standby database has the following features:

- It is a physical copy of a database, known as the primary database. Data files, archived logs, and control files are transferred from the primary database and applied to the standby database to keep the two databases synchronized. The databases should reside on physically different machines to protect against hardware failure.
- The standby database is constantly kept mounted and in Recovery mode, because archived log files are frequently transferred from the primary database. This process can be automated by the DBA.
- The primary database can be operated in Managed Recovery mode. In this mode, Oracle RDBMS keeps the standby database synchronized with the primary database by transferring archived logs from primary database to standby and applying them at standby automatically.

**Standby Database Features (continued)**

- A standby database can be opened in Read Only mode using the ALTER DATABASE OPEN READ ONLY command.

Queries can be run on the database in Read Only mode, but DML activity is not possible. It can then be returned to Recovery mode again by shutting down the database and mounting it using the STARTUP NOMOUNT and ALTER DATABASE MOUNT STANDBY DATABASE commands.
- You can transfer all archived logs to the standby machine and apply them by recovering the standby database.
  - If the current log cannot be transferred before the standby database is activated, then the standby database cannot be brought to the current time of failure and data may be lost.
  - It is important to apply all redo logs to the standby database before activation, because the logs cannot be applied after activation of the standby.
- The standby database becomes a primary database after activation. All users now connect to the new primary database, while the problem causing failure on the other machine is corrected. The failed database can then be set up as the new standby database by creating a new standby control file.

## Standby Database Guidelines

### Guidelines

- The standby database operates on Oracle release 7.3 or later.
- Standby and primary databases should be on different machines.
- The same version, release, and patch of the operating systems and RDBMS are on the primary and standby hosts.  
The standby host may use a different disk configuration
- Standby can be activated (READONLY) and returned to the Standby mode.
- During the activated period, archive logs cannot be applied

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Standby Database Guidelines

To use a standby database, you need to be aware of the following guidelines:

- The standby database command set was introduced from Oracle version 7.3. Versions prior to 7.3 cannot use this feature.
- The standby concept is most effectively implemented to protect against hardware failure. The standby and primary databases should therefore reside on separate machines.
- The primary and standby databases must have the same version, release, and patch of the Oracle server and operating system, because the process involves transferring physical files between two machines.

It is recommended, but not required, that the database identification string be the same, and the data, redo log, and control files have the same names on both the primary and standby databases.



---

## Initialization Parameters

### Initialization Parameters

The following parameters are significant:

- **COMPATIBLE**
- **DB\_FILES**
- **CONTROL\_FILES**
- **DB\_FILE\_NAME\_CONVERT**
- **LOG\_FILE\_NAME\_CONVERT**
- **STANDBY\_ARCHIVE\_DEST**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Initialization Parameters

Most parameters for your primary and standby database should be identical.

Initialization parameters, such as **CONTROL\_FILES**, **LOG\_FILE\_NAME\_CONVERT**, and **DB\_FILE\_NAME\_CONVERT** should be changed. The following parameters are important:

- **COMPATIBLE**: This parameter must be the same in both databases.
- **DB\_FILES**: This parameter in conjunction control file parameter should be identical to allow a same number of files on both databases.
- **CONTROL\_FILES**: The values should be different to reduce the chances of accidentally overwriting the other database's control files (especially if you are considering standby on a cluster or the same machine).
- **DB\_FILE\_NAME\_CONVERT** and **LOG\_FILE\_NAME\_CONVERT**: Use these parameters to convert the location of data files and log files if the directory names are different between the standby and primary machines.
- **STANDBY\_ARCHIVE\_DEST**: This parameter is used solely by the standby RFS process to determine the directory in which to place the archived logs. Oracle uses this value along with **LOG\_ARCHIVE\_FORMAT** to generate the log filename for the standby site.

## Creating a Standby Database

### Creating a Standby Database

**Process for creating a standby database:**

- **Back up data files for the primary database.**
- **Create the standby database control file.**
- **Archive the primary database's current online redo logs.**
- **Transfer the archived logs, data files, and control file to the standby machine.**
- **Mount the standby database.**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE<sup>®</sup>

### How to Create a Standby Database

- 1 Copy the `init.ora` file from the primary server to the standby database. Edit the following `init.ora` parameters for the standby database:

`CONTROL_FILES`, `DB_FILE_NAME_CONVERT`, and  
`LOG_FILE_NAME_CONVERT`

You may also need to change parameters such as `LOG_ARCHIVE_DEST`, `LOG_ARCHIVE_FORMAT`, `BACKGROUND_DUMP_DEST`, and `USER_DUMP_DEST` which provide the file locations to reflect the file and directory layout on the standby.

- 2 Create the control file for the standby database by issuing the following command on the primary database

```
SQL> alter database create standby controlfile as <filename>;
```

This command creates a copy of the primary database's control file. Copy the control file specified above to the standby server at the appropriate location specified by the `CONTROL_FILES` parameter of standby database.

- 3 Back up the data files belonging to the primary database. You can take the backups while the database is online or offline. You can use the online tablespace backup methodology if you are using online backup. Otherwise, shut down the primary database cleanly and back up data files. Use this backup and copy them to the appropriate location in the standby database server.

**How to Create a Standby Database (continued)**

- 4** If you have used online backup in step 3, archive the current online logs of the primary database by issuing:

```
SQL> alter system archive log current;
```

This forces a log switch to archive the current log to guarantee complete recovery can be performed on the standby database. Copy the archive logs to the standby database server.

- 5** Start the standby database in Nomount mode.

```
SQL> startup pfile=$HOME/initsndby.ora nomount
```

- 6** Mount the database using the following command:

```
SQL> alter database mount standby database;
```

- 7** Issue the recover command and apply the archived logs to the standby database to synchronize it with the primary database by using a special recover command:

```
SQL> recover standby database;
```

**Note**

- The standby database must be kept in Standby Recovery mode. This allows Oracle processes to recover the database by applying archived logs to the data files and prevents the database being accessed by any users.
- Once the standby database is activated to Normal mode, it *cannot* be returned to standby recovery mode without re-creating it as a standby database.

## Managed Recovery Mode

### Managed Recovery Mode

- The standby database automatically applies the archived redo logs when the files become available.
- Manual or automated transfer of archived redo logs is done by the DBA.

```
STARTUP NOMOUNT pfile=initSTANDBY.ora
ALTER DATABASE MOUNT STANDBY DATABASE;
RECOVER MANAGED STANDBY DATABASE TIMEOUT 60
```

- The recovery operation waits the specified number of minutes for applying archive logs.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Standby Database in Managed Recovery Mode

In Managed Recovery mode, a standby database waits for archived log files from a target database and then automatically applies the redo logs when the files become available. This feature eliminates the need for you to interactively provide the recovery process with filenames of the archived redo logs. The task of copying archive logs should be done by the DBA and can be automated separately.

### How to Place the Standby Database in Managed Recovery Mode

- 1 Use SQL\*Plus to start the standby database without mounting it.

```
STARTUP NOMOUNT pfile=initSTANDBY.ora
```

- 2 Mount the database.

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```

- 3 Put the standby database in managed recovery mode.

```
RECOVER MANAGED STANDBY DATABASE TIMEOUT 60;
```

By using the TIMEOUT option of the RECOVER statement, you can specify a timeout interval (in minutes). In this case, the managed recovery operation waits the specified number of minutes for the Oracle server to write the requested archived log entry to the standby control file's directory.

## Maintaining the Standby Database

### Maintaining the Standby Database

The primary and standby databases should be kept synchronized by:

- Regularly transferring archived logs from the primary database to the standby database
- Applying the archived logs to the standby database through recovery
- Avoiding commands that invalidate the standby database

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE™

### Maintaining the Standby Database

The standby database should frequently be synchronized with the primary database. This is achieved through the following process:

- Regularly transfer archived logs from the primary database to the standby database. The frequency will depend on how fast the standby database is required after the primary database fails.
- After the archived logs have been transferred, use the recover command to apply them to the standby database.
- Avoid incomplete recovery and clearing unarchived log files on the primary database, because these will invalidate the standby database. If such activities are required, then the standby database may have to be re-created.
- When operations involving NOLOGGING option are performed on the primary database, you have to take a backup of the tablespace involved and copy it to the standby database.

**Note:** Do not transfer redo logs. Transferring redo logs from the primary database will cause an error.

## Read Only Mode of Standby Database

### Read Only Mode of Standby

- Allow users to query an open database
- No DML operations
- Add tempfile entries in read-only mode for queries
- Unavailable for managed recovery

#### In manual recovery mode:

```
RECOVER CANCEL;
ALTER DATABASE READ ONLY;
```

#### In managed recovery mode:

```
RECOVER MANAGED STANDBY DATABASE CANCEL;
ALTER DATABASE OPEN READ ONLY;
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE<sup>®</sup>

### Opening the Standby Database in Read Only Mode

The Read Only mode allows users to query an open database, eliminating the potential for online data modifications. This functionality enables you to use your standby database as a temporary reporting database. Temporary tablespaces allow you to add tempfile entries in Read Only mode for the purposes of making queries.

If you maintain your standby database primarily for disaster prevention, you may be forced to activate the standby database immediately after a disaster. Furthermore, using a standby database for queries makes it unavailable for managed recovery and limits the standby's role as a disaster recovery database.

### How to Open the Standby Database in Read Only Mode in Manual Recovery

- 1 Cancel the recovery.

```
RECOVER CANCEL
```

- 2 Open the database in Read Only mode.

```
ALTER DATABASE OPEN READ ONLY;
```

## **How to Open the Standby Database in Read Only Mode When in Managed Recovery**

- 1 Start a SQL\*Plus session and execute the following:

```
RECOVER MANAGED STANDBY DATABASE CANCEL;
```

- 2 Open the database in Read Only mode.

```
ALTER DATABASE OPEN READ ONLY;
```

## **How to Move the Standby Database from Read Only Mode Back to Manual Recovery Mode**

- 1 Terminate all active user sessions on the standby database.
- 2 Issue the following statement:

```
RECOVER STANDBY DATABASE
```

You can also set the TIMEOUT option in the managed option.

## Activating the Standby Database

### Activating the Standby Database

If the primary database fails, the standby database can be used immediately by:

- Archiving the primary database's current online redo logs (if possible)
- Transferring and applying the archives to the standby database
- Activating the standby database
- Allowing future user connections to be established on the standby database

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### How to Activate the Standby Database

In the event of a disaster on the primary machine, you need to make the standby database active. To achieve this successfully, follow these steps:

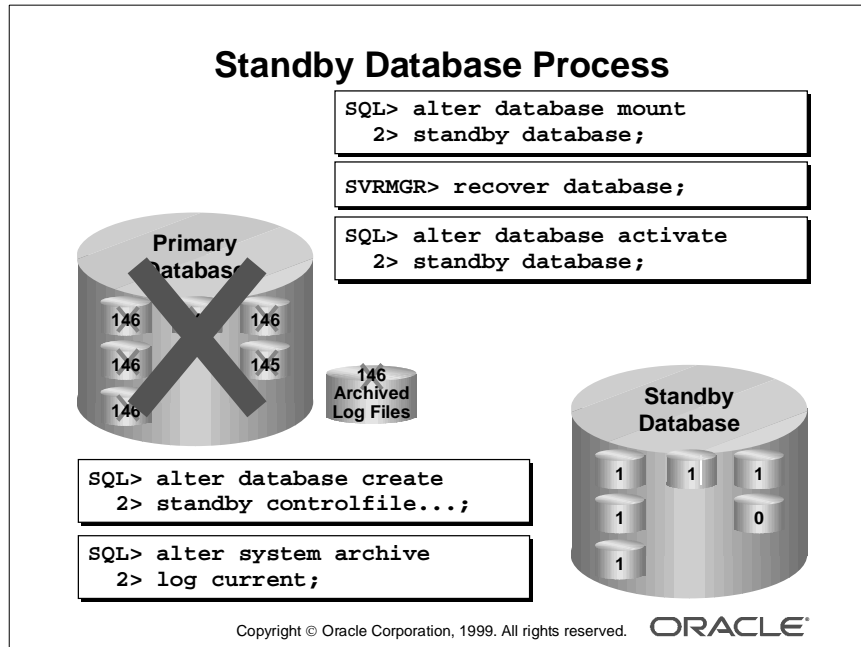
- 1 If possible, archive the current redo logs and transfer them to the standby machine.
- 2 Ensure that the standby database is mounted in Exclusive mode.
- 3 Apply the primary database's archived logs by recovering the standby database:  
`SQL> recover standby database;`
- 4 Cancel the recovery and activate the standby database. In issuing the `ALTER DATABASE ACTIVATE STANDBY DATABASE` command, the standby bit in the control file is reset, and takes the database back to Nomount mode:  
`SQL> alter database activate standby database;`
- 5 Shut down the standby instance and backup if possible. Then open the database:  
`SQL> alter database mount;`  
`SQL> alter database open;`  
Users can now connect to the new primary database.



### **How to Activate the Standby Database (continued)**

Apply all primary database archived logs to the standby database before activation, because the online logs of the standby database are reset during this process. Copying the online logs, or logs from the primary database that are not yet archived, to the standby database will not be usable by the standby database and will result in error.

## Operating a Standby Database



### Operating a Standby Database

The process of creating and activating a standby database is described below:

- 1 The primary database is operating normally, with current log sequence number 144. To create a standby database, create a standby control file and a current backup of the data files from the primary database.
- 2 Archive the current online redo log on the primary database using the ALTER SYSTEM command. This increments the log sequence number and produces archive 144.
- 3 Transfer the data files, standby control file, `init.ora` file, and all archived redo logs since the data file backup from the primary machine to the standby machine.
- 4 Start up the standby database in Nomount mode, then use the ALTER DATABASE command to mount the standby database.

While this process is occurring, users are working on the primary database and new archives have been generated on the primary database. The standby database is not fully synchronized with the primary database.

---

**Operating a Standby Database (continued)**

- 5** To synchronize the standby database, transfer the new archives from the primary database and recover the standby database.

While the users are working, a power surge destroys three disks on the primary machine. The primary database has crashed and cannot be opened without reorganizing space on disk, which will take too long. It is decided to activate the standby database.

- 6** First, take all files in the primary database that were damaged during the power surge offline. The system and rollback segment tablespaces, control file, and `init.ora` are all available. Open the database in Restricted mode, and archive the current online redo logs, because all other logs have been archived earlier. Then shut down the primary database.

```
SQL> startup mount pfile=$HOME/initDB00.ora restrict;
SQL> alter database datafile <damaged file> offline immediate;
SQL> alter database open;
SQL> alter system archive log current;
SQL> shutdown immediate
```

In some cases it may not be possible to open the primary database and hence you may not be able to obtain the current log file. In such cases, you may lose data (transactions) that were posted after the last archive log available.

- 7** Transfer the archived logs to the standby database. Recover the standby database, which is now identical to the primary database before it crashed.

```
SQL> recover standby database;
ORA-00279: change 309012...12/09/97 11:33:56 needed for thread 1
ORA-00289: suggestion : /disk1/archive/arch_146.rdo
ORA-00280: change 309012 for thread 1 is in sequence #146
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
cancel
Media recovery cancelled.
```

- 8** Activate the standby database using the `ALTER DATABASE ACTIVATE` command. This causes the standby database to perform a `resetlogs`. The standby database is your new primary database. Take a cold backup if there is time. Then mount and open the database.
- 9** The old primary machine can now be fixed and becomes the new standby machine for a new standby database.

## Structural Change of Primary Database

### Altering the Physical Structure at the Primary Database

Update the standby database's control file.

- **Adding data files**
  - Copy the data file from the primary to the standby database.
  - Add the datafile to the standby database with the **ALTER DATABASE CREATE DATAFILE** command.
  - Re-create the control file for the standby database.
- **Altering log files**
  - Adding or dropping log file groups does not affect the standby database.
  - Clearing log files on the primary database invalidates the standby database.

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Altering the Physical Structure at the Primary Database

Altering the physical structure of the primary database requires that the standby database's control file be updated to keep the databases synchronized.

#### Adding Data Files

- Adding a datafile to the primary database requires adding the same data file to the standby database.
- The **DB\_FILE\_NAME\_CONVERT** initialization parameter converts the name of the added data file.

When the redo stream detects the added data file, the recovery process attempts to open and write to it. If the data file cannot be found or opened on the standby database, the recovery process halts with an error.

#### How to Add Data Files

- 1** Copy the datafile from the primary to the standby database.
- 2** Add the datafile to the standby database with the **ALTER DATABASE CREATE DATAFILE** command.
- 3** Re-create the control file for the standby database.

## **How to Add Data Files (continued)**

**Altering Log Files** Adding or dropping log file groups or members from the primary database can be done without affecting the standby database.

Clear log files on the primary database by issuing the `ALTER DATABASE CLEAR UNARCHIVED LOGFILE` command or using the `RESETLOGS` option when opening the database or backing up the control file invalidates the standby database.

## Refreshing the Control File

### How to Refresh the Standby Database Control File

1. Halt the recovery process at the standby database.
2. Re-create the standby control file from the primary database.
3. Archive the current online redo logs from the primary database.
4. Transfer the control file and archived logs.
5. Mount the standby database and restart the recovery process.

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### How to Refresh the Control File

The standby database may need to be refreshed or re-created when the control file on the primary database is changed. For example, a CREATE CONTROLFILE command is used to increase the maximum number of redo log groups.

The procedure to refresh control file is listed below:

- 1 Create the control file for the standby database at the primary database:

```
SQL> alter database create standby controlfile
2> as <standby controlfile name>;
```

- 2 Archive the current online logs of the primary database.

- 3 Halt the recovery process on the standby database.

```
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
cancel
```

- 4 Transfer the standby control file, archived log files, and a copy of any data files to the standby database site.

- 5 Restart the recovery process on the standby database.

```
SQL> recover standby database;
```

## Nologging Operations at the Primary Database

### Nologging Operations

- **Data from nologging operations at primary are not recorded in log files. Propagate these as follows:**
  - Take a copy of all the data files comprising the tablespace involved.
  - Take the affected data files offline at the standby database and drop the tablespace.
  - Transfer the data file copies and current archived logs from the primary database to the standby database and perform recovery at standby.
- **Offline tablespaces**
  - Are not recovered
  - May be unusable when the standby database is activated

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE

### Nologging Operations at the Primary Database

When data loading operations involving the NOLOGGING option are used in the primary database, such data cannot be recovered from the redo logs. To propagate these changes, perform one of the following tasks:

- Take an online (or offline) copy of all the data files comprising the tablespace involved in NOLOGGING operation at the primary database.
- Take the affected data files offline at the standby database and drop the tablespace.
- Transfer the affected data files and current archived logs from the primary database to the standby database and perform recovery at standby.

### Standby Database File Options

Data files in the standby database can be altered offline as a mechanism to support a subset of the primary database's data files on the standby database.

Data files that are offline in the standby database are not recovered and are unusable when the standby database is activated. The files are logically unavailable, as if they had been dropped from the database.

## Summary

### Summary

**In this lesson, you should have learned that:**

- **A standby database is used where down time must be minimized**
- **The standby feature comes with Oracle 7.3 and later**
- **Keep directory structures for the primary and standby machines the same**
- **The standby and primary databases must reside on different machines**
- **A standby database cannot be used until activated**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**



---

## Quick Reference

| Context                   | Reference                                                                                                                                                                                                                                                                                                                        |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameters                | DB_FILES<br>DB_FILE_NAME_CONVERT<br>COMPATIBLE<br>CONTROL_FILES<br>LOG_FILE_NAME_CONVERT                                                                                                                                                                                                                                         |
| Dynamic performance views | None                                                                                                                                                                                                                                                                                                                             |
| Data Dictionary Views     | None                                                                                                                                                                                                                                                                                                                             |
| Commands                  | ALTER DATABASE ACTIVATE STANDBY DATABASE;<br>DATABASE CREATE STANDBY CONTROLFILE AS ...;<br>ALTER DATABASE DATAFILE <name> OFFLINE;<br>ALTER DATABASE MOUNT STANDBY DATABASE;<br>ALTER DATABASE OPEN RESTRICT;<br>ALTER DATABASE RECOVER STANDBY DATABASE ...;<br>ALTER SYSTEM ARCHIVE LOG CURRENT;<br>RECOVER STANDBY DATABASE; |



---

15

---

## Workshop

## Objectives

### Objectives

**After completing this lesson, you should be able to do the following:**

- **Document a database configuration using the Database Configuration Worksheet**
- **Configure an Oracle8 database to support stated business requirements**
- **Perform a full offline database backup**
- **Recover a failed database while minimizing down time and data loss**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Instructor Note

Carefully facilitate each session to ensure that timing is maintained. Emphasize the objectives and timing to students at the beginning of each session. It is up to the instructor to select from the list of scenarios based on student interests. The workshop is structured to address six scenarios over the course of the day. Additional scenarios can be performed as warranted by student interest and knowledge level.

## Workshop Methodology

### Workshop Methodology

- **Group-oriented and interactive**
- **Intensive hands-on diagnosis and problem resolution**
- **Variety of failure scenarios**
- **Recovery solutions**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE<sup>®</sup>

### Group-Oriented and Interactive Structure

The workshop is structured to allow individuals to work in groups to perform database backup, restore, and recovery operations. Each group is encouraged to share their approach to resolving database failures with other groups in the class.

### Intensive Hands-On Diagnosis and Problem Resolution

The intent is to provide you with as much hands-on experience as possible to diagnose and work through backup and recovery scenarios. Experience and knowledge gained from the first two days of the *Oracle8i: Backup and Recovery* course will play a major role toward successfully completing the objectives of each session.

## **Variety of Failure Scenarios**

During this workshop, the instructor will induce failure scenarios without informing the class. The objective is to diagnose the nature of the failure and to perform the appropriate recovery process. The types of failures that you may encounter include:

- Loss of a redo log group
- Loss of data files
- Loss of control files
- Media loss
- Loss of a table
- Loss of a primary database
- Corrupted data block
- Loss of an online rollback segment data file
- Time-based and change-based recovery

## **Recovery Solutions**

This workshop simulates a real-world environment in that exact solutions to problems may not be readily available in the event of a database failure. Therefore, only cursory instructions are noted in Appendix D for performing restore and recovery operations. In some cases you may use associated Oracle Worldwide Support bulletins included in Appendix E. The intent is to familiarize you with available documentation and how to interpret it to perform a successful database recovery.

## Workshop Approach

### Workshop Approach

- **Physical Investigation:**
  - Use views and tools to derive information
  - View trace files and log files
- **Database Configuration:**
  - Archiving is enabled
  - Control files and log files are mirrored
  - Control file is backed up

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE<sup>®</sup>

### Physical Investigation

Use the tools in the Oracle8i environment, such as the V\$ views, data dictionary views, and facilities in the Server Manager, and Oracle Enterprise Manager (if available) to derive information about your database environment. Keep the business requirements in mind and note any deficiencies that you feel will need to be corrected or implemented to support them.

### Database Configuration

Physically modify the database configuration to ensure that:

- Archiving is enabled
- Control files and log files are mirrored and distributed across multiple devices
- The control file is backed up both to trace and in a binary format

## Business Requirements

### Business Requirements

- **Twenty-four hour availability**
- **Peak usage varies across all time zones**
- **Daily backups are required**
- **Complete database recovery is required**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### Business Requirements

The following business requirements should be familiar to you when configuring your database for backup and recovery.

**Twenty-Four Hour Availability** The database must be available 24 hours a day, 7 days a week. An eight-hour maintenance window is scheduled for the first Saturday of each month when the instance can be shut down.

**Peak Usage Varies Across Available Time Frame** This database is accessed globally, so it is used throughout the 24-hour period of one day.

**Daily Backups** Full database backups are required on a daily basis.

**Complete Database Recovery** This is a critical business application and data loss cannot be tolerated. A high number of transactions occur over the 24-hour time frame.



## Resolving a Failure

### Resolving a Failure

- **Phase I: Diagnose the problem**
- **Phase II: Restore appropriate files**
- **Phase III: Recover the database**
- **Phase IV: Back up the database**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

### Resolving a Workshop Failure Scenario

The failure/recovery scenarios are hands-on exercises, where each group has the flexibility to perform the restore and recovery operation that they determine is appropriate. Approximately six failure/recovery scenarios will be conducted during the workshop. The workshop is structured for three scenarios before lunch and three scenarios after lunch, however the instructor can run additional scenarios if time allows.

The instructor will not tell you what scenario is being conducted. When resolving failures, follow these diagnostic steps:

### **Phase I: Diagnose the Problem**

- 1** The first phase is to research the nature of the failure. Use V\$ views, data dictionary views, trace and log files, basic operating system commands, and Oracle Enterprise Manager to diagnose the problem.
- 2** Determine if the database instance is available and the database is open.
- 3** Attempt to start the instance.
- 4** Shut down the instance if a problems occur while starting it or when opening the database.
- 5** Check the trace files and alert log file.
- 6** Determine the appropriate recovery method:
  - Closed database recovery
  - Open database, offline tablespace recovery
  - Open database, offline tablespace, individual data file recovery
  - Cancel-based recovery
  - Time-based recovery
  - Change-based recovery
- 7** Once you have determined what type of failure you are dealing with, refer to Appendix A for additional instructions and information.

### **Phase II: Restore Appropriate Files**

Before you perform a recovery scenario, determine what files to restore and what state the instance and database must be in to perform the recovery. Remember that the objective is to minimize down time and loss of data, so do not restore files if it is not necessary.

### **Phase III: Recover the Database**

Once the appropriate files are restored, initiate your recovery operation. After completing the recovery, note any proactive measures that can be taken to prevent that type of failure in the future.

### **Phase IV: Back Up the Database**

Not all recovery operations require a database backup when they are complete. However, determine whether your database needs to be backed up and, if so, perform another backup.

## Summary

### Summary

- **Instructor-facilitated workshop**
- **Group-oriented**
- **Hands-on approach**
- **Simulated “real-world” environment**
- **Minimize down time and data loss**
- **Use tools and diagnostics to determine the type of failure.**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

### **Instructor-Facilitated Workshop**

The instructor will facilitate the workshop by providing guidance and additional information as appropriate. This is a change from instructor-led training where the instructor does the majority of the talking.

### **Group-Oriented Emphasis**

A strong emphasis is placed on teaming with other students in the workshop for purposes of diagnosing and resolving failures. The ability to successfully complete each scenario is based upon the cumulative knowledge and problem resolution skills of each group.

### **Hands-On Approach**

This is meant to be a hands-on workshop, providing students with the maximum allowable time to be involved in a lab situation.

### **Simulated Real-World Environment**

Real-world problems are not as easy to resolve as hypothetical lab problems. Therefore, this workshop provides numerous database failures that will be unknown to you and must be diagnosed and resolved by each group using available resources and tools. The primary objective of each scenario is perform a timely recovery with no or minimal down time and data loss.

### **Minimized Down Time and Data Loss**

For many of the failure scenarios there may be more than one method to restore and recover the database. Keep in mind that when resolving the failure, you want to minimize down time and data loss, so select the best restore and recovery method.

### **Use of Tools and Diagnostics**

Use the various tools and diagnostics that were addressed during the first two days of the course to help determine and resolve the type of failure.

---

A

---

**Practices**

## **Practice Session Note**

Make sure that your instructor provides you with the proper connect descriptor for both your lab database and the Recovery Catalog database. The Practice Session solutions assume a connect descriptor for the lab database to be db01 and the connect descriptor for the Recovery Catalog to be db16. These may be different, however, for your installation.

## Practice 2-1

### Oracle Recovery Structures and Processes

- 1 What is the V\$ view that you must query for the names of all data files in the database?
- 2 What are the V\$ views that you must query for finding the current online redo log and names of all redo logs in the database?
- 3 What is the V\$ view that you must query for the names of all control files in the database?
- 4 Name the V\$ view that you should check to find the name of the database before dropping tables or shutting down the database.
- 5 Name the V\$ view that you must query to locate processes still connected to the instance before shutting down the database.
- 6 Describe the significance of the parameters LOG\_CHECKPOINT\_INTERVAL and FAST\_START\_IO\_TARGET in instance recovery.
- 7 What `init.ora` parameter configures the memory area in the SGA that buffers recovery information in memory before being written to disk?
- 8 What is the large pool, when is it used, and what initialization parameter configures it?
- 9 Optional: Set up mirroring of control files so as to have three control files.
- 10 Optional: Set up mirroring of redo log files so as to have three members per group.

## Practice 3-1

### Oracle Backup and Recovery Configuration

- 1 Connect to SQL\*Plus as `sysdba` and mount the database.
- 2 List the parameters `LOG_ARCHIVE_DEST`, `LOG_ARCHIVE_START`, and `LOG_ARCHIVE_FORMAT`, and note the values.
- 3 Execute the command `ARCHIVE LOG LIST`. Note the log mode of the database and whether automatic archival is enabled.
- 4 Set the database in `archivelog` mode.
- 5 Open the database.
- 6 Shut down the instance with the `IMMEDIATE` option.
- 7 Edit the `init.ora` file to:
  - Enable archiving
  - Archive log files to two destinations: `$HOME/ARCHIVE` and `$HOME/ARCHIVE2` directories (The `$HOME/ARCHIVE` is mandatory, and `$HOME/ARCHIVE2` is optional.)
  - Use the archiving format of `arch_%s.arc`
  - Spawn two archive processes at instance start
- 8 While editing the `init.ora` file, uncomment the `ROLLBACK_SEGMENTS` parameter to place the rollback segments `RBS01` and `RBS02` when the database starts.
- 9 Start up and open the database.
- 10 Verify that two archive processes are running.
- 11 Execute the `ALTER SYSTEM SWITCH LOGFILE` command twice, then show the values of the `ARCHIVE` parameters. Do you see any archived log files? What is the format of the filename?

### Advanced Practice

- 1 Stop automatic archiving by executing the `ALTER SYSTEM ARCHIVE LOG STOP` command.
- 2 Execute the `ALTER SYSTEM SWITCH LOGFILE` command enough times to recycle through all the online redo log groups. What happened and why?
- 3 Establish a second telnet session and connect to SQL\*Plus as `sysdba`.
- 4 Enable automatic archiving by using the `ALTER SYSTEM ARCHIVE LOG START` command.
- 5 Switch to your first session. What happened and why?



## Practice 4-1

### Closed Database Backup

- 1 While the database is open, connect to the database as `sys` or `system` and using `V$` and Data Dictionary Views, make a list of all of the files that must be backed up for a whole offline database backup.

**Note:** Copy the redo logs for ease of restore/recovery in `noarchivelog` mode.

- 2 Shut down the database with the `IMMEDIATE` option. Make two whole offline database backups using the operating system commands. Place one in `$HOME/DONTOUCH` directory, and the other in `$HOME/BACKUP`.

**Note:** Do not place or remove files from the `DONTOUCH` directory without instructor supervision. This copy will be used as the base in Scenarios.

- 3 Start the instance.
- 4 Connect as `scott/tiger` and execute the `$HOME/LABS/deptdata.sql` and `$HOME/LABS/empdata.sql` scripts to create transactions against the database.
- 5 Connect as `system/manager` and make an open backup of the `DATA01` tablespace. Copy the file to `$HOME/BACKUP` directory. Make sure that you do not overwrite another copy.
- 6 Use the `ALTER DATABASE` command to back up the control file to trace. Connect to Server Manager and execute the `$HOME/LABS/spid.sql` script to identify the trace file. Host to the operating system and copy the trace file to `$HOME/BACKUP/cntrl.sql`. Using an editor, remove the comments from the SQL file.
- 7 Create a binary copy of the control file and put it in the `$HOME/BACKUP` directory. Name the backup copy `cntrl1.bkp`.

## Practice 5-1

### Complete Database Recovery: Noarchivelog Mode

- 1 Shut down the database and disable automatic archival (ARC0) by editing the `init.ora` parameter file, then start and mount the database. Set the database in noarchivelog mode, and then open the database. Ensure the desired status by issuing `ARCHIVE LOG LIST` command.
- 2 Shut down the database and perform a full, closed backup by using the operating system command. (Copy the files to the `$HOME/BACKUP/NOARC` directory only.) Start the instance and open the database.
- 3 Start SQL\*Plus and connect by using `scott/tiger` and run the `$HOME/LABS/newemp.sql` script. This script creates new tables in scott's schema and adds some information to them.
- 4 Connect as `system/manager` and run the following script to record the names of data files that contain the table `NEWEMP`:

```
SQL> select f.file_name from dba_tables t, dba_data_files f
 2 where table_name = 'NEWEMP' and
 3 t.tablespace_name=f.tablespace_name;
 $HOME/DATA/DISK3/data01.dbf
```
- 5 Run the `$HOME/LABS/breakdb.sql` as `sys` or `system` in SQL\*Plus to simulate failure.
- 6 Attempt to restart the database normally. What happened?
- 7 Shut down the database and use the appropriate operating system command to replace the current database with the latest backup (from `NOARC` directory to `DATA` directory).
- 8 Start up and open the database so that it will be available to all users.
- 9 Connect to the database as `scott/tiger` to execute a query against the `NEWEMP` table. What happened and why?
- 10 What conclusions can you make about offline backups and recovery for databases in noarchivelog mode?

---

## Practice 5-2

### Complete Database Recovery: Archivelog Mode

- 1 Query the V\$DATABASE view to see what mode the database is in. Use ARCHIVE LOG LIST to check the status of archivelog mode and automatic archiving.
- 2 Shut down the instance and uncomment the LOG\_ARCHIVE\_START parameter in the init.ora. Mount the database and use the ALTER DATABASE command to enable archiving.
- 3 Check to make sure archiving is successful by using the ARCHIVE LOG LIST command. Note the current log sequence number.
- 4 Perform a closed database backup. Store the backup in the \$HOME/BACKUP directory.
- 5 Start SQL\*Plus and connect by using scott/tiger and run the \$HOME/LABS/newemp.sql script. This script simulates the creation of a new table and users adding important information to it.
- 6 Connect as system/manager and run the script \$HOME/LABS/checktbs.sql to note the data files associated with the tablespace that contains the table NEWEMP.
- 7 Run the \$HOME/LABS/breakdb.sql script to simulate hardware failure.
- 8 Attempt to shut down and restart the database normally. What happened?
- 9 The database cannot locate the files for the DATA01 tablespace because of perceived media failure. Since archiving is enabled, you can now perform a complete recovery.
- 10 Use the RECOVER DATABASE command to recover the database.
- 11 When recovery is complete, open the database to make it available for all users.
- 12 Query the DBA\_TABLESPACES view to see if the tablespace DATA01 is online.
- 13 Connect to the database as scott/tiger and execute a query against the NEWEMP table. What happened?
- 14 Connect as system/manager and query the V\$LOG view and note the sequence number. Compare this value with the value in step 3. What conclusions can you make about complete recovery?

### Optional Practice 1

- 1 Run the `$HOME/LABS/breakdb.sql` script to simulate hardware failure.
- 2 Attempt to restart the database normally. What happened?
- 3 The database cannot locate the files for the `USER_DATA` tablespace because of a perceived media failure. Because archiving is enabled, you can now perform a complete recovery.
- 4 Take the data files for the `DATA01` tablespace offline.
- 5 Open the database to make it available for all users.
- 6 Take the `DATA01` tablespace offline, then restore all data files from the backup.
- 7 Use the `RECOVER TABLESPACE` command to recover the tablespace.
- 8 Put the `USER_DATA` tablespace back online.
- 9 Connect to the database as `scott/tiger` and execute a query against the `NEWEMP` table to make sure it still exists.

### Optional Practice 2

- 1 Run the `$HOME/LABS/newtbs.sql` script to
  - Create a new tablespace with a new data file
  - Create a table `NEW_EMP1` with data on the new tablespace
  - Simulate the loss of the new data file

The database cannot locate the file for the `NEW_DATA` tablespace because of perceived loss of file. Since archiving is enabled, you can perform a complete recovery after the recreation of the file for which you have no backup.

- 2 You can either take the data file for the `NEW_DATA` tablespace offline, or take the tablespace offline, because it only contains one data file.

**Note:** `IMMEDIATE` option must be included to avoid a checkpoint trying to write to a nonexistent file.

Confirm the recovery status by querying `V$RECOVER_FILE` to check the status of a backup.

- 3 You now must re-create the file.
- 4 Use the `RECOVER` or `ALTER DATABASE RECOVER` commands to start applying the archives and the redo logs to the recreated data file.
- 5 To bring the data file to the point of failure, all needed archived logs and redo logs are applied.

- 6** All data files are now synchronized. When recovery is finished, bring the tablespace online.  
All data is now recovered. Include the file in the backup strategy and notify users that the tablespace is ready to be used again.
- 7** Connect to the database as `scott/tiger` and execute a query against the `NEW_EMP1` table to make sure it still exists.

### **Optional Practice 3**

While you perform the online backup of the DATA01 tablespace, simulate a shutdown abort of your database. You will need to recover the situation to reopen the database to the users though an online backup was being performed and not finished.

- 1** Make an online backup of the DATA01 tablespace.
- 2** Make an OS backup of the tablespace files.
- 3** Shut down in ABORT mode.
- 4** Mount the database.
- 5** Retrieve from the appropriate dictionary view the information on the active online backed up data file to be ended.
- 6** Unfreeze the header to release the backup mode on the data file.
- 7** Open the database for users.

---

## Practice 6-1

### Recovering from User Failure: Incomplete Recovery

- 1 If you are unsure whether you have a valid backup from the previous exercise, then Perform either a whole closed or opened database backup. Store the backup in the `$HOME/BACKUP` directory.
- 2 Start SQL\*Plus and connect using `scott/tiger` and insert rows into the NEWEMP table by executing the following statement:  

```
SQL> insert into newemp select * from newemp;
```
- 3 Select a count of the rows in the NEWEMP table. Note the number of rows.
- 4 Connect as `system/manager` and record the data files belonging to the tablespace that contains the NEWEMP table.
- 5 Record the current system time using an operating system command.
- 6 Record the current online log sequence number by querying V\$LOG.
- 7 Connect using `scott/tiger` and add rows to the NEWEMP table by executing the following command:  

```
insert into newemp select * from newemp;
```
- 8 Get a count of the rows in the NEWEMP table.
- 9 Run the `$HOME/LABS/breaktab.sql` script to simulate a user accidentally dropping the NEWEMP table.  

```
SQL> @$HOME/LABS/breaktab.sql
```
- 10 Attempt to query the NEWEMP table. What happened?
- 11 The database cannot locate the NEWEMP table. Mount the database to perform recovery.
- 12 Restore all data files from the backup that you made in step 1.  
**Note:** Check the V\$DATAFILE view with the `$HOME/LABS/cpdbfile.sh` script. If the files and locations are identical, then run this script. Otherwise, you may modify this script to match your database structure.
- 13 Recover the database until the time you noted in step 4.
- 14 When recovery is complete, open the database using the RESETLOGS option to enable access for all users.
- 15 Connect to the database as `scott/tiger` and execute a query against the NEWEMP table. What happened and why?
- 16 Connect as `system/manager`, query the V\$LOG view, and note the sequence number. Compare this value with the value in step 5. What conclusions can you make about incomplete recovery?
- 17 Take a whole offline backup. Store the backup in the `$HOME/BACKUP` directory.

### Advanced Practice 6-1: Recovery with a Lost Archived Log: Incomplete Recovery

- 1 Perform a complete closed backup.
- 2 Record the current system time using an operating system command.
- 3 Record the current online log sequence number by querying the V\$LOG view.
- 4 Run the `$HOME/LABS/breakarc.sql` script as user SYSTEM to simulate the loss of an archived log file.  
`SQL> @$HOME/LABS/breakarc.sql`
- 5 Run the `$HOME/LABS/breakdb.sql` script to simulate hardware failure.
- 6 Attempt to restart the database normally. What happened?
- 7 The database cannot locate the files for the DATA01 tablespace because of perceived media failure. Since archiving is enabled, you can attempt to perform a complete recovery.
- 8 Use the RECOVER AUTOMATIC DATABASE command to recover the database, and note the archived log name that cannot be found. Issue a CANCEL when the database is unable to locate the specified archive log.
- 9 Attempt to open the database. What happened?
- 10 The recovery has been cancelled prior to applying the lost archived log. The data files in the DATA01 tablespace, therefore, cannot be brought forward to the current database time. Since recovery cannot take the database back in time, you must perform an incomplete recovery.
- 11 Recover the database using the UNTIL CANCEL option, stopping before the Oracle server requests the archived log file you noted in step 8.  
**Note:** Do not use the automatic method. Apply each archived log manually as the Oracle server requests it.
- 12 Type `cancel` at the recovery prompt.
- 13 Once recovery is complete, open the database using the RESETLOGS option to enable access for all users.
- 14 Check to make sure all data files are online, then shut down and take a full offline backup.



## Practice 7-1

### Oracle Export and Import Utilities

Use the Export and Import utilities to save flat file copies of data offline.

- 1** Invoke the Export utility to export the EMP and DEPT tables in the scott schema.
- 2** With SQL\*Plus, connect as `scott`. Drop the EMP and DEPT tables.
- 3** Restore the EMP and DEPT tables by using the import utility.
- 4** Query the EMP and DEPT tables to get the number of rows in each of those tables.

## Practice 8-1

### Starting the Oracle Database with a Missing Data File

- 1 Connect as sysdba.
- 2 Shut down the database by using IMMEDIATE.
- 3 From the operating system prompt, rename the data file `$HOME/DATA/DISK3/temp01.dbf` to `$HOME/DATA/DISK3/temp01.old`.
- 4 Connect as sysdba and start the database. What error message do you receive?
- 5 Drop the data files for the TEMPORARY tablespace.
- 6 Open the database.
- 7 Drop the TEMPORARY tablespace.
- 8 Re-create the TEMPORARY tablespace by using the filename `temp01.dbf` with a size of 50K.

### Reconstruct Lost Control Files

- 1 Connect as sysdba.
- 2 Create a trace file of the control file by using the ALTER DATABASE command. Edit the trace file to remove the comments so that you will end up with only a script of appropriate SQL\*Plus commands needed for re-creating the control file. Rename this file to `$HOME/BACKUP/crecntl.sql`.
- 3 Shut down the database by using IMMEDIATE.
- 4 From the operating system prompt, remove the control files `$HOME/DATA/DISK1/cntrl_1.ctl` and `$HOME/DATA/DISK2/cntrl_2.ctl`.
- 5 Connect as sysdba and start the database. What error message do you receive?
- 6 Execute the `$HOME/BACKUP/crecntl.sql` trace file script to re-create the control files.
- 7 Shutdown the database. If needed, create a new password file by using the password file utility and a password of `oracle`.
- 8 Start up the database to ensure that your database is functioning correctly.

### Recover a Read-Only Tablespace

- 1 Connect as sysdba.
- 2 Shut down the database by using IMMEDIATE.
- 3 From the operating system prompt, rename the data file `$HOME/DATA/DISK1/query01.dbf` to `$HOME/DATA/DISK1/query01.old`.
- 4 Connect as sysdba and start the database. What error message do you receive?
- 5 Restore the data files for the QUERY\_DATA tablespace.  
`$ mv query01.old query01.dbf`
- 6 Open the database.

## Practice 9-1

### Types of Troubleshooting

- 1 Execute the ALTER SYSTEM SWITCH LOGFILE command twice. Change to the BDUMP directory, list the files, then view the last few lines of alert\_<SID>.log file. You notice that Log Switch is recorded in alert log.
- 2 Connect to SQL\*Plus and use the SHOW command to check the values of the parameter DB\_BLOCK\_CHECKSUM.
- 3 Run the DBVERIFY utility against the user01.dbf data file. Do you see any corrupted blocks?

### Use of LogMiner Utility

- 1 A new employee Scott inserts a new row into SCOTT.EMP table. Use the values clause (7777, 'HENRY', 'CLERK', 7566, '19-APR-87', 9999, 0, 30).
- 2 The next day, the DBA is asked to cancel the wrong insertion made some time the previous day by Scott on SCOTT.EMP table. The user, however, cannot remember which row was inserted.

Initialize the LogMiner utility to undo the SQL insert statement executed by Scott on SCOTT.EMP table the previous day.

- 3 Specify redo log files to be analyzed.
- 4 Start LogMiner analysis.
- 5 Find the insertion made the day before (in reality, during the last five minutes). Use the following script.

```
SQL> select username, timestamp, sql_redo, sql_undo
2 from v$logmnr_contents
3 where seg_name = 'EMP' and seg_owner = 'SCOTT'
4 and username = 'SCOTT' and timestamp = '<DD-MON-YY>';
```

- 6 Cancel the insertion.
- 7 Stop LogMiner analysis.

## Practice 10-1

- 1 List some benefits that your organization could obtain by using RMAN.

---

---

---

---

---

- 2 What is a recovery catalog, when should it be used, and where should it be created?

---

---

---

---

---

---

- 3 List the benefits of recovery catalog.

---

---

- 4 While the instance is running, connect to your target database in RMAN without using the recovery catalog.

- 5 What are stored scripts? How are they useful?

---

---

---

---

---

## Practice 11-1

This practice involves learning how to manage, maintain, set up scripts in, and query the recovery catalog for Recovery Manager. Remember that you have two databases for this practice: your lab account database—target database, and the Recovery Catalog database, which contains a recovery catalog schema for your student account.

- 1 Connect to your target database and recovery catalog using RMAN.
- 2 Execute the command to resync the control file and recovery catalog. What happened? Why?
- 3 Register target database in the recovery catalog at the RMAN prompt.
- 4 Using RMAN, list all the database incarnations registered in the catalog.
- 5 Enter the RESET DATABASE command at the RMAN prompt. What happens?
- 6 View the script `$HOME/LABS/crebkup.sql` script, and then run it as user `system` by logging into target database using `SQL*PLUS` to create an online operating system copy of the system tablespace data file in your `$HOME/BACKUP` directory.
- 7 Using RMAN, add the backup mode in step 6 to the catalog.
- 8 Using RMAN, confirm that the data file has been added to the recovery catalog.
- 9 Use the RMAN command to remove the backup of the system data file from the recovery catalog. *Do not* remove the file from the disk.
- 10 Use the REPORT command to determine which data files have not yet been backed up by RMAN today.
- 11 Connect to your recovery catalog database using `SQL*Plus`, and query `RC_DATAFILE_COPY` to confirm that the data file has been removed from the recovery catalog.
- 12 Create a script to make a whole database backup with following information:  
 Name of script:nightback  
 Channel name: dbnD (*n* is the student account number)  
 Channel type Disk  
 Format `$HOME/BACKUP/%b%d%s%p`  
 Level Database (No archive logs)  
 tag nback  
 DO NOT RUN THIS SCRIPT NOW.
- 13 Query the recovery catalog and verify whether the script has been created.

## Practice 12-1

### Backup Using RMAN

- 1 What are the two supported backup types for Recovery Manager? List some of the differences between the two backup types.
- 2 Create an image copy of data files belonging to SYSTEM tablespace. The copy should be placed in \$HOME/BACKUP/INC0 directory with the name of sys0101.cpy. The tag should be SYSTEM01.
- 3 Create a script to back up the database following these guidelines:
  - The name of the script should be FULLBACK.
  - Make a backup set.
  - The incremental level should be 0.
  - Backup pieces should be on disk in \$HOME/INC0 directory.
  - The format string should include FULL%n\_%s.%p.
  - Do not include archive logs.
  - Place four files per backup set.
  - Parallel the backup operation in three sessions.
- 4 Run the script created in step 3 and verify completion of the backup.
- 5 Using RMAN, back up archive logs generated today to \$HOME/BACKUP/INC1 directory.
- 6 Log in to target database using SQL\*PLUS as user scott, and run the script moreemp.sql. Create a backup of DATA01 tablespace with the following guidelines:
  - It should be a backup set.
  - The incremental level should be 2.
  - Do not include control file.
  - Backup piece should be on disk in \$HOME/INC2 directory.
- 7 Obtain a listing of all data files that have not been backed up in the last two days.

## Practice 13-1

### Restore and Recover Using RMAN

It is assumed that the previous practice in Lesson 12 has been completed successfully.

- 1** Obtain a list of backup sets registered in the catalog.
- 2** Obtain a list of copies listed in the catalog.
- 3** Run the script `breakdb.sql` as `sysdba` user in a `SQL*PLUS` session. Then, using RMAN, recover the `DATA01` tablespace.
- 4** You have determined that `DISK3 ($HOME/DATA/DISK3)` is corrupted. You must relocate all the files on `DISK3` to another location. You determined that `DISK4` has sufficient space. Using RMAN, relocate all the data files in `DISK3` to `DISK4`.





---

B

---

## **Practice Solutions**

## **Practice Session Note**

Make sure that your instructor provides you with the proper connect descriptor for both your lab database and the Recovery Catalog database. The practice session solutions assume a connect descriptor for the lab database to be U7 and the connect descriptor for the Recovery Catalog to be T3. These may be different, however, for your installation.

## Practice 2-1 Solutions

### Oracle Recovery Structures and Processes

- 1 What is the V\$ view that you must query for the names of all data files in the database?  
**V\$DATAFILE**
- 2 What are the V\$ views that you must query for finding the current online redo log and names of all redo logs in the database?  
**V\$LOG and V\$LOGFILE**
- 3 What is the V\$ view that you must query for the names of all control files in the database?  
**V\$CONTROLFILE**
- 4 Name the V\$ view that you should check to find the name of the database before dropping tables or shutting down the database.  
**V\$DATABASE**
- 5 Name the V\$ view that you must query to locate processes still connected to the instance before shutting down the database.  
**V\$PROCESS**
- 6 Describe the significance of the parameters LOG\_CHECKPOINT\_INTERVAL and FAST\_START\_IO\_TARGET in instance recovery.  
**When the LOG\_CHECKPOINT\_INTERVAL parameter is set, the target for checkpoint position cannot lag the end of the log by more than the number of redo log blocks specified by this parameter. This ensures that no more than a fixed number of redo blocks will need to be read during instance recovery.**  
**The dynamic initialization parameter FAST\_START\_IO\_TARGET allows you to limit the number of blocks that must be read for crash or instance recovery. If the value of this parameter is smaller, the recovery performance is better because fewer blocks need to be recovered.**
- 7 What init.ora parameter configures the memory area in the SGA that buffers recovery information in memory before being written to disk?  
**LOG\_BUFFER**
- 8 What is the large pool, when is it used, and what initialization parameter configures it?  
**The large pool is an area of the SGA which can be used for buffering information in memory for Recovery Manager when IO slaves are required. This increases the speed and efficiency of backups and restores when using RMAN.**  
**The LARGE\_POOL\_SIZE parameter specifies the number of bytes allocated from the SGA.**

- 9** Optional: Set up mirroring of control files so as to have three control files.

**To add a new control file or change the number or location of the control file, use the following steps:**

- a** Shut down the database:

```
SQL> SHUTDOWN NORMAL
```

- b** Copy the existing control file to a different device using operating system commands:

```
$cp -p $HOME/DATA/DISK1/control01.ctl $HOME/DATA/DISK3/
control03.ctl
```

```
$chmod g+wx $HOME/DATA/DISK3/control03.ctl
```

- c** Edit or add the **CONTROL\_FILES** parameter and specify names for all the control files:

```
$vi initdb1.ora
control_files=$HOME/DATA/DISK1/control01.ctl,
 $HOME/DATA/DISK2/control02.ctl,
 $HOME/DATA/DISK3/control03.ctl
```

- d** Start up the database:

```
SQL> STARTUP PFILE=$HOME/LABS/initU7.ora
```

- 10** Optional: Set up mirroring of redo log files so as to have three members per group.

```
SQL> alter database add logfile member
2 '$HOME/DATA/DISK3/log0103.log' to group 1,
3 '$HOME/DATA/DISK3/log0203.log' to group 2;
```

## Practice 3-1 Solutions

### Oracle Backup and Recovery Configuration

- 1 Connect to SQL\*Plus as sysdba and place the database in mount state.

**If the database is open, then shutdown the database. Exit the SQL\*PLUS session, then follow these steps:**

```
$ sqlplus /nolog
```

```
SQL> connect / as sysdba
```

```
SQL> startup mount pfile=$HOME/LABS/initU7.ora
```

```
ORACLE instance started.
```

- 2 List the parameters LOG\_ARCHIVE\_DEST, LOG\_ARCHIVE\_START, and LOG\_ARCHIVE\_FORMAT, and note the values.

```
SQL> show parameter archive
```

| NAME                         | TYPE    | VALUE       |
|------------------------------|---------|-------------|
| log_archive_dest             | string  | ~/dbs/arch  |
| log_archive_dest_1           | string  |             |
| log_archive_dest_2           | string  |             |
| log_archive_dest_3           | string  |             |
| log_archive_dest_4           | string  |             |
| log_archive_dest_5           | string  |             |
| log_archive_dest_state_1     | string  | enable      |
| log_archive_dest_state_2     | string  | enable      |
| log_archive_dest_state_3     | string  | enable      |
| log_archive_dest_state_4     | string  | enable      |
| log_archive_dest_state_5     | string  | enable      |
| log_archive_duplex_dest      | string  |             |
| log_archive_format           | string  | %t_%s.dbf   |
| log_archive_max_processes    | integer | 1           |
| log_archive_min_succeed_dest | integer | 1           |
| log_archive_start            | boolean | FALSE       |
| standby_archive_dest         | string  | ~/dbs/arch) |

- 3 Execute the command ARCHIVE LOG LIST. Note the log mode of the database and whether automatic archival is enabled.

```
SQL> archive log list;

Database log mode No Archive Mode
Automatic archival Disabled
Archive destination /oracle/core/8.1.5/dbs/arch
Oldest online log sequence 240
Next log sequence to archive 241
Current log sequence 241
```

- 4 Set the database in archivelog mode.

```
SQL> alter database archivelog;

Database altered.
```

- 5 Open the database.

```
SQL> alter database open;

Database altered.
```

- 6 Shut down the instance with the IMMEDIATE option.

```
SQL> shutdown immediate;

Database closed.
Database dismounted.
ORACLE instance shut down.
```

- 7 Edit the `init.ora` file to:

- Enable archiving
- Archive log files to two destinations: `$HOME/ARCHIVE` and `$HOME/ARCHIVE2` directories (The `$HOME/ARCHIVE` is mandatory, and `$HOME/ARCHIVE2` is optional.)
- Use the archiving format of `arch_%s.arc`
- Spawn two archive processes at instance start

`init.ora` file

...

```
log_archive_start = true
log_archive_dest_1 = "LOCATION=$HOME/ARCHIVE/ MANDATORY"
log_archive_dest_2 = "LOCATION=$HOME/ARCHIVE2/ OPTIONAL"
log_archive_max_processes = 2
log_archive_format = arch_%s.arc
```

- 8 While editing the `init.ora` file, uncomment the `ROLLBACK_SEGMENTS` parameter to place the rollback segments `RBS01` and `RBS02` when the database starts.

```
rollback_segments=(rbs01, rbs02)
```

- 9 Start up and open the database.

```
SQL> startup pfile=$HOME/LABS/initU7.ora
```

- 10 Verify that two archive processes are running.

```
$ ps -ef|grep arc
```

```
oracle 29296 1 0 03:19:51 ? 0:00 ora_arc0_db01
oracle 29298 1 0 03:19:51 ? 0:00 ora_arc1_db01
```

- 11 Execute the `ALTER SYSTEM SWITCH LOGFILE` command twice, then show the values of the `ARCHIVE` parameters. Do you see any archived log files? What is the format of the filename?

```
SQL> alter system switch logfile;
```

```
System altered.
```

```
SQL> alter system switch logfile;
```

```
System altered.
```

```
SQL> select name, value
```

```
2 from v$parameter
```

```
3 where name like 'log_archive%';
```

| NAME                     | VALUE                             |
|--------------------------|-----------------------------------|
| log_archive_start        | TRUE                              |
| log_archive_dest         |                                   |
| log_archive_duplex_dest  |                                   |
| log_archive_dest_1       | LOCATION=\$HOME/ARCHIVE MANDATORY |
| log_archive_dest_2       | LOCATION=\$HOME/ARCHIVE2 OPTIONAL |
| log_archive_dest_3       |                                   |
| log_archive_dest_4       |                                   |
| log_archive_dest_5       |                                   |
| log_archive_dest_state_1 | enable                            |
| log_archive_dest_state_2 | enable                            |
| log_archive_dest_state_3 | enable                            |
| log_archive_dest_state_4 | enable                            |
| log_archive_dest_state_5 | enable                            |

```

log_archive_max_processes 2
log_archive_min_succeed_dest 1
log_archive_format arch_%s.arc
SQL> !ls -l $HOME/ARCHIVE $HOME/ARCHIVE2
/db/oracle/archive/:
total 154
-rw-rw---- 1 oracle dba 77824 Mar 23 03:28 arch_121.arc
-rw-rw---- 1 oracle dba 1024 Mar 23 03:28 arch_122.arc
/db/oracle/archive2/:
total 154
-rw-rw---- 1 oracle dba 77824 Mar 23 03:28 arch_121.arc
-rw-rw---- 1 oracle dba 1024 Mar 23 03:28 arch_122.arc

```

### Advanced Practice

- 1 Stop automatic archiving by executing the ALTER SYSTEM ARCHIVE LOG STOP command.  

```
SQL> alter system archive log stop;
```

System altered.
- 2 Execute the ALTER SYSTEM SWITCH LOGFILE command enough times to recycle through all the online redo log groups. What happened and why?  

```
SQL> alter system switch logfile;
```

System altered.

```
SQL> alter system switch logfile;
```

**The database is in archivelog mode, but since automatic archiving is disabled the next redo log file cannot be used since it has not been archived.**
- 3 Establish a second telnet session and connect to SQL\*Plus as sysdba.
- 4 Enable automatic archiving by using the ALTER SYSTEM ARCHIVE LOG START command.  

```
SQL> alter system archive log start;
```

System altered.
- 5 Switch to your first session. What happened and why?  

**You now have the message “System Altered” followed by the SQL prompt. The archive process has been restarted. The log switch is now possible because the redo log files were archived when archiving was restarted.**



## Practice 4-1 Solutions

### Closed Database Backup

- 1 While the database is open, connect to the database as `sys` or `system` and using `V$` and Data Dictionary Views, make a list of all of the files that must be backed up for a whole offline database backup.

**Note:** Copy the redo logs for ease of restore/recovery in `noarchivelog` mode.

```
SQL> select name from v$controlfile;
```

NAME

```

/u03/user/db02/DATA/DISK1/control01.ctl
/u03/user/db02/DATA/DISK2/control02.ctl
```

```
SQL> select member from v$logfile;
```

MEMBER

```

/u03/user/db02/DATA/DISK3/redo01a.log
/u03/user/db02/DATA/DISK4/redo01b.log
/u03/user/db02/DATA/DISK4/redo02a.log
/u03/user/db02/DATA/DISK3/redo02b.log
```

```
SQL> select name from v$datafile;
```

NAME

```

/u03/user/db02/DATA/DISK1/system01.dbf
/u03/user/db02/DATA/DISK2/rbs01.dbf
/u03/user/db02/DATA/DISK3/data01.dbf
/u03/user/db02/DATA/DISK2/temp01.dbf
/u03/user/db02/DATA/DISK2/indx01.dbf
/u03/user/db02/DATA/DISK3/oemrep01.dbf
/u03/user/db02/DATA/DISK1/query01.dbf
```

```
7 rows selected.
```

- 2 Shut down the database with the IMMEDIATE option. Make two whole offline database backups using the operating system commands. Place one in \$HOME/DONTOUCH directory, and the other in \$HOME/BACKUP.

```
$ cp -rp $HOME/DATA/* $HOME/DONTOUCH
$ cp $ORACLE_HOME/dbs/orapw* $HOME/DONTOUCH
$ cp -rp $HOME/DATA/* $HOME/BACKUP
$ cp $ORACLE_HOME/dbs/orapw* $HOME/BACKUP
```

**Note:** Do not place or remove files from the DONTOUCH directory without instructor supervision. This copy will be used as the base in Scenarios.

- 3 Start the instance.
- 4 Connect as scott/tiger and execute the \$HOME/LABS/deptdata.sql and \$HOME/LABS/empdata.sql scripts to create transactions against the database.

```
SQL> connect scott/tiger
SQL> @$HOME/LABS/empdata.sql;
SQL> @$HOME/LABS/deptdata.sql;
```

- 5 Connect as system/manager and make an open backup of the DATA01 tablespace. Copy the file to \$HOME/BACKUP directory. Make sure that you do not overwrite another copy.

```
SQL> connect system/manager
SQL> alter tablespace data01 begin backup;
SQL> !cp $HOME/DATA/DISK3/data01.dbf $HOME/BACKUP/data01.bkp
SQL> alter tablespace data01 end backup;
```

- 6 Use the ALTER DATABASE command to back up the control file to trace. Connect to SQL\*PLUS and execute \$HOME/LABS/spid.sql script to identify the trace file. Exit to the operating system and copy the trace file to \$HOME/BACKUP/cntrl.sql. Using an editor, remove the comments from the SQL file.

```
SQL> alter database backup controlfile to trace;
SQL> @$HOME/LABS/spid.sql
```

| USERNAME | SPID         |
|----------|--------------|
| -----    | -----        |
| SYSTEM   | <process ID> |

```
SQL> exit
$ cd $HOME/UDUMP
$ cp U1_ora_<process ID>.trc $HOME/BACKUP/cntrl.trc
$ vi $HOME/BACKUP/cntrl.trc
<Remove all comments from the trace file.
Add the PFILE keyword to the STARTUP command.>
```

- 7** Create a binary copy of the control file and put it in the \$HOME/BACKUP directory. Name the backup copy cntrl1.bkp.

```
SQL> alter database backup controlfile to '$HOME/BACKUP/
cntrl1.bkp';
```

## Practice 5-1 Solutions

### Complete Database Recovery: Noarchivelog Mode

- 1 Shut down the database and disable automatic archival (ARC0) by editing the `init.ora` parameter file, then start and mount the database. Set the database in noarchivelog mode, and then open the database. Ensure the desired status by issuing `ARCHIVE LOG LIST` command.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
$ vi $HOME/LABS/initdb01.ora
 comment out the log_archive_start parameter
$ sqlplus /nolog
SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup mount pfile=$HOME/LABS/initU7.ora
SQL> alter database noarchivelog;
sql> alter database open;
sql> archive log list;
Database log mode Noarchive Mode
Automatic archival Disabled
Archive destination $HOME/ARCHIVE/
Oldest online log sequence 243
Next log sequence to archive 244
Current log sequence 244
```

- 2 Shut down the database and perform a full closed backup by using the operating system command. (Copy the files to the `$HOME/BACKUP/NOARC` directory only.) Start the instance and open the database.

```
SQL> shutdown immediate
SQL> exit
$ cp -r $HOME/DATA/* $HOME/BACKUP/NOARC
$ sqlplus /nolog
SQL> connect / as sysdba
SQL> startup pfile=$HOME/LABS/initU7.ora
```

- 
- 3 Start SQL\*Plus and connect by using `scott/tiger` and run the `$HOME/LABS/newemp.sql` script. This script creates new tables in `scott`'s schema and adds some information to them.
- ```
SQL> @$HOME/LABS/newemp.sql
```
- 4 Connect as `system/manager` and run the following script to record the names of data files that contain the table `NEWEMP`:
- ```
SQL> select f.file_name from dba_tables t, dba_data_files f
 2 where table_name = 'NEWEMP' and
 3 t.tablespace_name=f.tablespace_name;
 $HOME/DATA/DISK3/data01.dbf
```
- 5 Run the `$HOME/LABS/breakdb.sql` as `sys` or `system` in SQL\*Plus to simulate failure.
- ```
SQL> @$HOME/LABS/breakdb.sql
```
- 6 Attempt to restart the database normally. What happened?
- ```
ORA-01157: cannot identify data file 3- file not found
ORA-01110: data file 3: '$HOME/DATA/DISK3/data01.dbf'
```
- The database cannot open data file number 4. Therefore, the database is left in the mount state. The database cannot locate the files for the DATA01 tablespace because of perceived media failure.**
- 7 Shut down the database and use the appropriate operating system command to replace the current database with the latest backup (from `NOARC` directory to `DATA` directory).
- ```
SQL> connect / as sysdba;
SQL> shutdown abort;
SQL> !cp -rp $HOME/BACKUP/NOARC/* $HOME/DATA
```
- 8 Start up and open the database so that it will be available to all users.
- ```
SQL> connect / as sysdba;
SQL> startup pfile=$HOME/LABS/initU7.ora
```
- 9 Connect to the database as `scott/tiger` to execute a query against the `NEWEMP` table. What happened and why?
- ```
SQL> connect scott/tiger;
SQL> select * from NEWEMP;
ORA-00942: table or view does not exist
```
- The table does not exist because it was created after the last backup was taken.**
- 10 What conclusions can you make about offline backups and recovery for databases in `noarchivelog` mode?
- Offline backups can be used to restore the database. Databases in `noarchivelog` mode have no archived logs to recover from the previous backup to the point of failure. Therefore, all changes after the previous backup have been lost. This explains why the `NEWEMP` table no longer exists.**
-

Practice 5-2 Solutions

Complete Database Recovery: Archivelog Mode

- 1 Query the V\$DATABASE view to see what mode the database is in. Use ARCHIVE LOG LIST to check the status of archivelog mode and automatic archiving.

```
SQL> select dbid, name, log_mode from v$database;
```

DBID	NAME	LOG_MODE
-----	----	-----
1943591421	DB01	NOARCHIVELOG ...

```
1 row selected.
```

```
SQL> archive log list;
```

Database log mode	Noarchive Mode
Automatic archival	disabled
Archive destination	\$HOME/ARCHIVE/
Oldest online log sequence	243
Next log sequence to archive	244
Current log sequence	244

- 2 Shut down the instance and uncomment the LOG_ARCHIVE_START parameter in the init.ora file. Mount the database and use the ALTER DATABASE command to enable archiving.

```
SQL> shutdown immediate;
```

```
SQL> exit
```

```
$ sqlplus /nolog
```

```
SQL> connect / as sysdba
```

```
SQL> startup mount pfile=$HOME/LABS/initU7.ora
```

```
SQL> alter database archivelog;
```

```
SQL> alter database open;
```

- 3 Check to make sure archiving is successful by using the ARCHIVE LOG LIST command. Note the current log sequence number.

```
SQL> archive log list;
```

Database log mode	Archive Mode
Automatic archival	Enabled
Archive destination	\$HOME/ARCHIVE/
Oldest online log sequence	243
Next log sequence to archive	244
Current log sequence	244

-
- 4 Perform a closed database backup. Store the backup in the \$HOME/BACKUP directory.
- ```
SQL> shutdown immediate;
SQL> !cp -rp $HOME/DATA/* $HOME/BACKUP
SQL> exit
$ sqlplus /nolog
SQL> connect / as sysdba
SQL > startup pfile=$HOME/LABS/initU7.ora
```
- 5 Start SQL\*Plus and connect by using scott/tiger and run the \$HOME/LABS/newemp.sql script. This script simulates the creation of a new table and users adding important information to it.
- ```
SQL> @$HOME/LABS/newemp.sql
```
- 6 Connect as system/manager and run the script \$HOME/LABS/checktbs.sql to note the data files associated with the tablespace that contains the table NEWEMP.
- ```
SQL> connect system/manager
SQL> @$HOME/LABS/checktbs
FILE_NAME

/u03/user/db02/DATA/DISK3/data01.dbf
1 row selected.
```
- 7 Run the \$HOME/LABS/breakdb.sql script to simulate hardware failure.
- ```
SQL> @$HOME/LABS/breakdb.sql
```
- 8 Attempt to shut down and restart the database normally. What happened?
- ```
ORA-01157: cannot identify data file 3- file not found
ORA-01110: data file 3: '$HOME/DATA/DISK3/data01.dbf'
```
- The database cannot open data file number 3. Therefore, the database is left in mount mode.**
- 9 The database cannot locate the files for the DATA01 tablespace because of perceived media failure. Since archiving is enabled, you can now perform a complete recovery.
- Restore the data files for the DATA01 tablespace from the backup that you made in step 4.**
- ```
$ cp -p $HOME/BACKUP/DISK3/data01.dbf $HOME/DATA/DISK3
```
- 10 Use the RECOVER DATABASE command to recover the database.
- ```
SQL> recover automatic database;
```

- 11** When recovery is complete, open the database to make it available for all users.

```
SQL> alter database open;
```

- 12** Query the DBA\_TABLESPACES view to see if the tablespace DATA01 is online.

```
SQL > select tablespace_name, status from dba_tablespaces
```

```
2> where tablespace_name = 'DATA01';
```

| TABLESPACE_NAME | STATUS |
|-----------------|--------|
| DATA01          | ONLINE |

1 row selected.

- 13** Connect to the database as scott/tiger and execute a query against the NEWEMP table. What happened?

```
SQL> select count(*) from newemp;
```

```
COUNT(*)
```

```

```

```
25
```

1 row selected

- 14** Connect as system/manager and query the V\$LOG view and note the sequence number. Compare this value with the value in step 3. What conclusions can you make about complete recovery?

| GROUP# | THREAD# | SEQUENCE# | BYTES  | MEMBERS | ARC | STATUS   |
|--------|---------|-----------|--------|---------|-----|----------|
| ...    |         |           |        |         |     |          |
| -----  | -----   | -----     | -----  | -----   | --- | -----    |
| ...    |         |           |        |         |     |          |
| 1      | 1       | 247       | 512000 | 2       | YES | INACTIVE |
| 2      | 1       | 248       | 512000 | 2       | NO  | CURRENT  |

The log sequence numbers now are higher than in step 3 where the database backup was taken after changing to archivelog mode. During recovery, archived logs have been applied, and the database has been brought forward to the current point in time.



## Optional Practice 1

- 1 Run the `$HOME/LABS/breakdb.sql` script to simulate hardware failure.

```
SQL> @$HOME/LABS/breakdb.sql
```

- 2 Attempt to restart the database normally. What happened?

```
ORA-01157: cannot identify data file 3- file not found
```

```
ORA-01110: data file 3: '$HOME/DATA/DISK3/data01.dbf'
```

**The database cannot open data file number 3. Therefore, the database is left in mount mode.**

- 3 The database cannot locate the files for the USER\_DATA tablespace because of a perceived media failure. Because archiving is enabled, you can now perform a complete recovery.

- 4 Take the data files for the DATA01 tablespace offline.

```
SQL > alter database datafile '$HOME/DATA/DISK3/data01.dbf'
 2 offline;
```

- 5 Open the database to make it available for all users.

```
SQL> alter database open;
```

- 6 Take the DATA01 tablespace offline, then restore all data files from the backup.

```
SQL> alter tablespace DATA01 offline immediate;
```

```
$ cp $HOME/BACKUP/DISK3/data01.dbf $HOME/DATA/DISK3
```

- 7 Use the RECOVER TABLESPACE command to recover the tablespace.

```
SQL> recover automatic tablespace DATA01;
```

- 8 Put the USER\_DATA tablespace back online.

```
SQL> alter tablespace DATA01 online;
```

- 9 Connect to the database as `scott/tiger` and execute a query against the NEWEMP table to make sure it still exists.

```
SQL> connect scott/tiger;
```

```
SQL> select count(*) from newemp;
```

```
COUNT(*)
```

```

```

```
S25
```

```
1 row selected
```

## Optional Practice 2

- 1 Run the `$HOME/LABS/newtbs.sql` script as user SYSTEM to:
  - Create a new tablespace with a new data file
  - Create a table NEW\_EMP1 in SCOTT's schema with data on the new tablespace
  - Simulate the loss of the new data file

```
SQL> @$HOME/LABS/newtbs.sql
```

The database cannot locate the file for the NEW\_DATA tablespace because of perceived loss of file. Since archiving is enabled, you can perform a complete recovery after the recreation of the file for which you have no backup.

- 2 You can either take the data file for the NEW\_DATA tablespace offline, or take the tablespace offline, because it only contains one data file.

**Note:** The immediate option must be included to avoid a checkpoint trying to write to a file which does not exist:

```
SQL > alter tablespace NEW_DATA offline immediate;
```

Tablespace altered.

Confirm the recovery status by querying V\$RECOVER\_FILE to check the status of a backup.

```
SQL> select * from v$recover_file;
```

| FILE# | ONLINE  | ERROR          | CHANGE# | TIME |
|-------|---------|----------------|---------|------|
| 7     | OFFLINE | FILE NOT FOUND | 0       |      |

- 3 You now must re-create the file.

```
SQL > alter database create datafile
```

```
2 '$HOME/DATA/DISK4/new01.dbf'
```

```
3 as '$HOME/DATA/DISK4/new01.dbf';
```

Database altered.

```
SQL> select * from v$recover_file;
```

| FILE# | ONLINE  | ERROR | CHANGE# | TIME      |
|-------|---------|-------|---------|-----------|
| 7     | OFFLINE |       | 248621  | 25-MAR-99 |

- 4 Use the RECOVER or ALTER DATABASE RECOVER commands to start applying the archives and the redo logs to the recreated data file.

```
SQL > recover tablespace NEW_DATA;
```

- 5** To bring the data file to the point of failure, all needed archived logs and redo logs are applied.

- 6** All data files are now synchronized. When recovery is finished, bring the tablespace online.

```
SQL > alter tablespace NEW_DATA online;
```

All data is now recovered. Include the file in the backup strategy and notify users that the tablespace is ready to be used again.

- 7** Connect to the database as `scott/tiger` and execute a query against the `NEW_EMP1` table to make sure it still exists.

```
SQL> connect scott/tiger;
```

```
SQL> select COUNT(*) from NEW_EMP1;
```

```
COUNT(*)
```

```

```

```
100
```

```
1 row selected
```

- 8** Drop the tablespace `NEW_DATA`.

```
SQL> drop tablespace new_data including contents;
```

### Optional Practice 3

While you perform the online backup of the DATA01 tablespace, simulate a shutdown abort of your database. You will need to recover the situation to reopen the database to the users, though an online backup was being performed and not finished.

- 1 Make an online backup of the DATA01 tablespace.

```
SQL> alter tablespace DATA01 begin backup;
Tablespace altered.
```

- 2 Make an OS backup of the tablespace files.

```
$ cp $HOME/DATA/DISK3/data01.dbf $HOME/BACKUP/DISK3
```

- 3 Shut down in ABORT mode.

```
SQL> shutdown abort;
```

- 4 Mount the database.

```
SQL> connect / as sysdba;
SQL> startup mount pfile=initdb01.ora;
```

- 5 Retrieve from the appropriate dictionary view the information on the active online backed up data file to be ended.

```
SQL> select * from v$backup;
```

| FILE# | STATUS     | CHANGE# | TIME      |
|-------|------------|---------|-----------|
| ----- | -----      | -----   | -----     |
| 1     | NOT ACTIVE | 0       |           |
| 2     | NOT ACTIVE | 0       |           |
| 3     | ACTIVE     | 228596  | 25-MAR-99 |
| 4     | NOT ACTIVE | 0       |           |
| 5     | NOT ACTIVE | 0       |           |
| 6     | NOT ACTIVE | 0       |           |
| 7     | NOT ACTIVE | 0       |           |

This indicates that file number 4 is currently in hot backup mode.

- 6 Unfreeze the header to release the backup mode on the data file.

```
SQL> alter database datafile 3 end backup;
Database altered.
```

```
SQL> select * from v$backup;
```

| FILE# | STATUS     | CHANGE# | TIME      |
|-------|------------|---------|-----------|
| ----- | -----      | -----   | -----     |
| 1     | NOT ACTIVE | 0       |           |
| ...   |            |         |           |
| 3     | NOT ACTIVE | 228596  | 25-MAR-99 |
| ...   |            |         |           |

- 7 Open the database for users.

```
SQL> alter database open;
Database altered.
```

## Practice 6-1 Solutions

### Recovering from User Failure: Incomplete Recovery

- 1 If you are unsure whether you have a valid backup from the previous exercise, then perform either a whole closed or opened database backup. Store the backup in the \$HOME/BACKUP directory.

```
SQL> shutdown immediate;
```

```
SQL> !cp -rp $HOME/DATA/* $HOME/BACKUP
```

- 2 Start SQL\*Plus and connect by using scott/tiger and insert rows into the NEWEMP table by executing the following statement:

```
SQL> insert into newemp select * from newemp;
```

- 3 Select a count of the rows in the NEWEMP table. Note the number of rows.

```
SQL> select count(*) from newemp;
```

```
COUNT(*)
```

```

```

```
78
```

- 4 Connect as system/manager and run the following commands:

```
SQL> select f.file_name from dba_tables t, dba_data_files f
```

```
2 where table_name = 'NEWEMP' and
```

```
3 t.tablespace_name=f.tablespace_name;
```

Record the filename of all data files for the tablespace.

```
$HOME/DATA/DISK3/data01.dbf
```

- 5 Record the current system time using an operating system command.

```
$ date
```

```
Wed Jun 09 17:18:04 PST 1999
```

- 6 Record the current online log sequence number by querying V\$LOG.

```
SQL> select * from v$log;
```

| GROUP# | THREAD# | SEQUENCE# | BYTES  | MEMBERS | ARC | STATUS   |
|--------|---------|-----------|--------|---------|-----|----------|
| ...    |         |           |        |         |     |          |
| -----  | -----   | -----     | -----  | -----   | --- | -----    |
| ...    |         |           |        |         |     |          |
| 1      | 1       | 251       | 512000 | 2       | YES | INACTIVE |
| 2      | 1       | 252       | 512000 | 2       | NO  | CURRENT  |

- 7 Connect using scott/tiger and add rows to the NEWEMP table by executing the following command:

```
$ sqlplus scott/tiger
```

```
SQL> insert into newemp select * from newemp;
```

- 8 Get a count of the rows in the NEWEMP table.

```
SQL> select count(*) from newemp;
COUNT(*)

 156
```

- 9 Run the \$HOME/LABS/breaktab.sql script to simulate a user accidentally dropping the NEWEMP table.

```
SQL> @$HOME/LABS/breaktab.sql
```

- 10 Attempt to query the NEWEMP table. What happened?

```
SQL> select * from scott.NEWEMP;
ORA-00942: table or view does not exist
```

**The table does not exist any more.**

- 11 The database cannot locate the NEWEMP table. Since archiving is enabled, you have all the archives, and you know the approximate time of failure, you can now perform an incomplete recovery to bring back the table.

**Shut down and then mount the database.**

```
SQL> shutdown immediate
SQL> startup mount pfile=$HOME/LABS/initU1.ora
```

- 12 Restore all data files from the backup that you made in step 1.

**Note:** Check the V\$DATAFILE view with the \$HOME/LABS/cpdbfile.sh script. If the files and locations are identical, then run this script. Otherwise, you may modify this file to match your database structure.

```
SQL> !cp $HOME/BACKUP/DISK1/<datafile1> \
2> $HOME/DATA/DISK1
SQL> !cp $HOME/BACKUP/DISK1/<datafile2> \
2> $HOME/DATA/DISK1
...
```

- 13 Recover the database until the time you noted in step 4.

```
SQL> recover database until time '1999-03-09:17:18:04'
```

- 14 When recovery is complete, open the database using the RESETLOGS option to enable access for all users.

```
SQL> alter database open resetlogs;
```

- 15 Connect to the database as `scott/tiger` and execute a query against the `NEWEMP` table. What happened and why?

```
SQL> connect scott/tiger
SQL> select count(*) from newemp;
COUNT(*)

 50
1 row selected
```

**The table now exists again, because the entire database is taken back to a time before the table was dropped. However, you lost the rows that were inserted after the time to which you recovered the database.**

- 16 Connect as `system/manager`, query the `V$LOG` view, and note the sequence number. Compare this value with the value in step 5. What conclusions can you make about incomplete recovery?

```
SQL> select * from v$log;
GROUP# THREAD# SEQUENCE# BYTES MEMBERS ARC STATUS
...

...
 1 1 1 512000 2 NO CURRENT
 2 1 0 512000 2 YES UNUSED
```

**The sequence numbers are reset to 0.**

- 17 Take a whole offline backup. Store the backup in the `$HOME/BACKUP` directory.

```
SQL> connect / as sysdba
SQL> shutdown immediate
SQL> !cp -rp $HOME/DATA/* $HOME/BACKUP
```

## Advanced Practice 6-1 Solution: Recovery with a Lost Archived Log: Incomplete Recovery

- 1 Perform a complete closed backup.
- 2 Record the current system time using an operating system command.
- 3 Start up and open the database if you have not already done so. Record the current online log sequence number by querying the V\$LOG view.

```
$ date
```

```
Wed Jul 22 17:18:04 PST 1999
```

```
SQL> select * from v$log;
```

| GROUP# | THREAD# | SEQUENCE# | BYTES  | MEMBERS | ARC | STATUS  |
|--------|---------|-----------|--------|---------|-----|---------|
| ...    |         |           |        |         |     |         |
| 1      | 1       | 1         | 512000 | 1       | NO  | CURRENT |
| 2      | 1       | 0         | 512000 | 0       | YES | UNUSED  |

- 4 Run the \$HOME/LABS/breakarc.sql script to simulate the loss of an archived log file.
- 5 Run the \$HOME/LABS/breakdb.sql script to simulate hardware failure.

```
SQL> @$HOME/LABS/breakarc.sql
```

```
SQL> @$HOME/LABS/breakdb.sql
```

- 6 Attempt to restart the database normally. What happened?

```
ORA-01157: cannot identify data file 3- file not found
```

```
ORA-01110: data file 4: '$HOME/DATA/DISK3/data01.dbf'
```

**The database cannot open data file number 3. Therefore, the database is left in mount mode.**

- 7 The database cannot locate the files for the DATA01 tablespace because of perceived media failure. Since archiving is enabled, you can attempt to perform a complete recovery.

**Restore the data files for the DATA01 tablespace from the backup you made in step 1.**

```
$ cp $HOME/BACKUP/DISK3/data01.dbf $HOME/DATA/DISK3
```



- 8 Use the RECOVER AUTOMATIC DATABASE command to recover the database, and note the archived log name that cannot be found. Issue a CANCEL when the database is unable to locate the specified archive log.

```
SQL> recover automatic database
ORA-00279: change 114408 generated ... needed for thread 1
ORA-00289: suggestion : $HOME/ARCHIVE/db01_<missing log>.arc
ORA-00280: change 114408 for thread 1 is in sequence #<mssing
log>
ORA-00278: log file ... no longer needed for this recovery
ORA-00308: cannot open ...'$HOME/ARCHIVE/db01_<missing log>.arc'
ORA-27037: unable to obtain file status
SVR4 Error: 2: No such file or directory
Additional information: 3
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
CANCEL
SQL>
```

- 9 Attempt to open the database. What happened?

```
SQL> alter database open;
ORA-01113: file 3 needs media recovery
ORA-01110: data file 3: '$HOME/DATA/DISK3/data01.dbf'
```

**The data file requires more recovery to become synchronized with the other data files.**

- 10 The recovery has been cancelled prior to applying the lost archived log. The data files in the DATA01 tablespace, therefore, cannot be brought forward to the current database time. Since recovery cannot take the database back in time, you must perform an incomplete recovery.

**Restore all data files from the backup you made in step 1.**

**Note:** Check the V\$DATAFILE view with the \$HOME/LABS/cpdbfile.sh script. If the files and locations are identical, then run this script. Otherwise, you can modify this file to match your database structure.

```
SQL> !$HOME/LABS/cpdbfile.sh
```

- 11** Recover the database using the UNTIL CANCEL option, stopping before the Oracle server requests the archived log file you noted in step 8.

**Note:** Do not use the automatic method. Apply each archived log manually as the Oracle server requests it.

```
SQL> recover database until cancel
```

```
...
```

Log applied.

```
ORA-00279: change 114408 ... needed for thread 1
```

```
ORA-00289: suggestion : $HOME/ARCHIVE/dbxx_<missing log>.arc
```

```
ORA-00280: change 114408 for thread 1 is in sequence #<missing log>
```

```
ORA-00278: log file ... no longer needed for this recovery
```

```
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
```

- 12** Type cancel at the recovery prompt.

```
cancel
```

Media recovery cancelled.

- 13** Once recovery is complete, open the database using the RESETLOGS option to enable access for all users.

```
SQL> alter database open resetlogs;
```

Statement processed.

- 14** Check to make sure all data files are online, then shut down and take a full offline backup.

```
SQL> select name, status from v$datafile;
```

| NAME                                   | STATUS |
|----------------------------------------|--------|
| /u03/user/db02/DATA/DISK1/system01.dbf | SYSTEM |
| /u03/user/db02/DATA/DISK2/rbs01.dbf    | ONLINE |
| /u03/user/db02/DATA/DISK3/data01.dbf   | ONLINE |
| /u03/user/db02/DATA/DISK2/temp01.dbf   | ONLINE |
| /u03/user/db02/DATA/DISK2/indx01.dbf   | ONLINE |
| /u03/user/db02/DATA/DISK3/oemrep01.dbf | ONLINE |
| /u03/user/db02/DATA/DISK1/query01.dbf  | ONLINE |

7 rows selected.

## Practice 7-1 Solutions

### Oracle Export and Import utilities

Use the Export and Import utilities to save flat file copies of data offline.

- 1 Invoke the Export utility to export the EMP and DEPT tables in the scott schema.

```
$ exp scott/tiger file=$HOME/export.dmp tables=emp,dept
```

- 2 With SQL\*Plus, connect as scott. Drop the EMP and DEPT tables.

```
$ sqlplus scott/tiger
```

```
SQL> drop table emp;
```

```
Table dropped.
```

- 3 Restore the EMP table by using the import utility.

```
$ imp scott/tiger file=$HOME/export.dmp TABLES=emp
```

- 4 Query the EMP and DEPT tables to find the number of rows in each of them.

```
SQL> select count(*) from emp;
```

```
COUNT(*)
```

```

```

```
25
```

```
SQL> select count(*) from dept;
```

```
COUNT(*)
```

```

```

```
12
```

## Practice 8-1 Solutions

### Starting the Oracle Database with a Missing Data File

- 1 Connect as sysdba.

```
SQL> connect / as sysdba
```

- 2 Shut down the database by using IMMEDIATE.

```
SQL> shutdown immediate
```

- 3 From the operating system prompt, rename the data file \$HOME/DATA/DISK3/temp01.dbf to \$HOME/DATA/DISK3/temp01.old.

```
$ cd $HOME/DATA/DISK2
```

```
$ mv temp01.dbf temp01.old
```

- 4 Connect as sysdba and start the database. What error message do you receive?

```
SQL> connect / as sysdba
```

```
SQL> startup open pfile=$HOME/LABS/initdb01.ora
```

```
ORA-01157: cannot identify/lock data file 4- see DBWR trace file
```

```
ORA-01110: data file 4: '$HOME/DATA/DISK2/temp01.dbf'
```

**Cannot open the database because a file is missing, so the database is left in mount mode.**

- 5 Drop the data files for the TEMPORARY tablespace.

```
SQL> alter database datafile '$HOME/DATA/DISK2/temp01.dbf'
```

```
2 offline drop;
```

```
Database altered.
```

- 6 Open the database.

```
SQL> alter database open;
```

```
Database altered.
```

- 7 Drop the TEMPORARY tablespace.

```
SQL> drop tablespace TEMP including contents;
```

- 8 Recreate the TEMPORARY tablespace using the filename temp01.dbf with a size of 50K.

```
SQL> create tablespace temp datafile
```

```
2 '$HOME/DATA/DISK2/temp01.dbf' size 50k;
```

---

## Reconstruct Lost Control Files

- 1 Connect as sysdba.  
`SQL> connect / as sysdba`
- 2 Create a trace file of the control file by using the ALTER DATABASE command. Edit the trace file to remove the comments so that you will end up with only a script of appropriate SQL\*Plus commands needed for recreating the control file. Rename this file to \$HOME/BACKUP/crecntl.sql.  
`SQL> alter database backup controlfile to trace;`  
`Database altered.`
- 3 Shut down the database by using IMMEDIATE.  
`SQL> shutdown immediate`  
`Database closed.`  
`Database dismounted.`  
`ORACLE instance shut down.`
- 4 From the operating system prompt, remove the control files \$HOME/DATA/DISK1/control01.ctl and \$HOME/DATA/DISK2/control02.ctl.  
`$ rm $HOME/DATA/DISK1/control01.ctl $HOME/DATA/DISK2/control02.ctl`
- 5 Connect as sysdba and start the database. What error message do you receive?  
`SQL> connect / as sysdba`  
`SQL> STARTUP`  
`ORA-00205 error in identifying controlfile, check alert log for more info`  
`SQL> SHUTDOWN`
- 6 Execute the \$HOME/BACKUP/crecntl.sql trace file script to re-create the control files.  
`SQL> $HOME/BACKUP/crecntl.sql`  
`Controlfile created.`  
`ORA-00283 recovery session canceled due to errors.`  
`ORA-00264 no recovery required.`  
`System altered.`  
`Database altered.`
- 7 Shut down the database. If needed, create a new password file using the password file utility and a password of oracle.
- 8 Start up the database to ensure that your database is functioning correctly.

## Recover a Read-Only Tablespace

- 1 Connect as sysdba.

```
SQL> connect / as sysdba
```

- 2 Shut down the database by using IMMEDIATE.

```
SQL> shutdown immediate
```

- 3 From the operating system prompt, rename the data file \$HOME/DATA/DISK1/query01.dbf to \$HOME/DATA/DISK1/query01.old.

```
$ cd $HOME/DATA/DISK1
```

```
$ mv query01.dbf query01.old
```

- 4 Connect as sysdba and start the database. What error message do you receive?

```
SQL> connect / as sysdba
```

```
SQL> startup open pfile=$HOME/LABS/initdb01.ora
```

```
ORA-01157 cannot identify/lock data file 6- see DBWR trace file
```

```
ORA-01110: data file 6: '$HOME/DATA/DISK1/query01.dbf'
```

**Cannot open the database because a file is missing, so the database is left in mount mode.**

- 5 Restore the data files for the QUERY\_DATA tablespace.

```
$ mv query01.old query01.dbf
```

- 6 Open the database.

```
SQL> alter database open;
```

```
Database altered.
```

## Practice 9-1 Solutions

### Types of Troubleshooting

- 1 Execute the ALTER SYSTEM SWITCH LOGFILE command twice. Change to the BDUMP directory, list the files, then view the last few lines of alert\_<SID>.log file. You notice that Log Switch is recorded in alert log.

```
SQL> alter system switch logfile;
```

```
SQL> alter system switch logfile;
```

```
SQL> !
```

```
$ cd $HOME/BDUMP
```

```
$ ls
```

```
$ vi alert_db01.log
```

```
(--> go to bottom of file)
```

- 2 Connect to SQL\*Plus and use the SHOW command to check the values of the parameter DB\_BLOCK\_CHECKSUM.

```
SQL> show parameter db_block_checksum
```

| NAME              | TYPE    | VALUE |
|-------------------|---------|-------|
| db_block_checksum | boolean | FALSE |

If you wanted to enable check summing for the online redo log files or database files, what would set this value to?

```
TRUE
```

- 3 Run the DBVERIFY utility against the user01.dbf data file. Do you see any corrupted blocks?

```
$ dbv file=$HOME/DATA/DISK3/data01.dbf
```

### Use of LogMiner Utility

- 1 A new employee Scott inserts a new row into SCOTT.EMP table.

```
SQL> connect SCOTT/TIGER
```

```
Connected.
```

```
SQL> insert into SCOTT.EMP values
```

```
2 (7777,'HENRY','HANK','HHENRY','19-APR-87','CLERK',
7566,'CLERK',30,9999,0);
```

```
1 row created.
```

```
SQL> COMMIT;
```

```
Commit complete.
```

- 2 The next day, the DBA is asked to cancel the wrong insertion made some time the previous day by Scott on SCOTT.EMP table. The user, however, cannot remember which row was inserted.

Initialize the LogMiner utility to undo the SQL insert statement executed by Scott on SCOTT.EMP table the previous day.

**Edit the init.ora and add the parameter:**

**UTL\_FILE\_DIR=**<full\_home\_path>/TRACE

**SQL> connect / as sysdba**

**Connected.**

**SQL> STARTUP**

**SQL> execute**

**DBMS\_LOGMNR\_D.build('logdict', '<full\_home\_path>/TRACE');**

**PL/SQL procedure successfully completed.**

- 3 Specify redo log files to be analyzed.

**SQL> execute dbms\_logmnr.add\_logfile('<full\_home\_path>/DATA/DISK3/redo0101.log', dbms\_logmnr.NEW);**

**PL/SQL procedure successfully completed.**

**SQL> execute dbms\_logmnr.add\_logfile('<full\_home\_path>/DATA/DISK4/redo0201.log', dbms\_logmnr.ADDFILE);**

**PL/SQL procedure successfully completed.**

- 4 Start LogMiner analysis.

**SQL> execute**

**dbms\_logmnr.start\_logmnr(dictfilename=>'<full\_home\_path>/TRACE/logdict');**

**PL/SQL procedure successfully completed.**

- 5 Find the insertion made the day before (in reality, during the last five minutes).

**SQL> select username, timestamp, sql\_redo, sql\_undo**

**2 from v\$logmnr\_contents**

**3 where seg\_name = 'EMP' and seg\_owner = 'SCOTT'**

**4 and username = 'SCOTT' and timestamp = '<DD-MON-YY>';**

**USERNAME**

**TIMESTAMP**

-----

**SQL\_REDO**

-----

-----

**SQL\_UNDO**



```


SYS 16-APR-99

insert into SCOTT.EMP(EMPNO,ENAME,...)
values (7777,'HENRY','CLERK',...);

delete from SCOTT.EMP where EMPNO = 7777 and ENAME = 'HENRY'
and ... and ROWID = 'AAACqqAACAAAegAAN';
6 Cancel the insertion.
SQL> connect SCOTT/TIGER
Connected.
SQL> delete from SCOTT.EMP where EMPNO = 7777
 2 and ENAME = 'HENRY' and JOB = 'CLERK' ...
 3 and ROWID = 'AAACqqAACAAAegAAN';
1 row deleted.
SQL> commit;
Commit complete.
7 Stop LogMiner analysis.
SQL> execute dbms_logmnr.end_logmnr;
PL/SQL procedure successfully completed.
```

## **Practice 10-1 Solutions**

There are no formal solutions to the questions in this practice.

## Practice 11-1 Solutions

This practice involves learning how to manage, maintain, set up scripts in, and query the recovery catalog for Recovery Manager. Remember that you have two databases for this practice: your lab account database—target database, and the Recovery Catalog database, which contains a recovery catalog schema for your student account.

- 1 Connect to your target database and recovery catalog using RMAN.

```
rman target / catalog db01/db01@db16
```

- 2 Execute the command to resync the control file and recovery catalog. What happened? Why?

```
RMAN> resync catalog;
```

```
RMAN-03022: compiling command: resync
```

```
RMAN-03023: executing command: resync
```

```
RMAN-08029: snapshot controlfile name set to default value: ?/
dbs/snapcf_@.f
```

```
RMAN-03026: error recovery releasing channel resources
```

```
RMAN-00571:
```

```
=====
```

```
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS=====
```

```
RMAN-00571:
```

```
=====
```

```
RMAN-03006: non-retryable error occurred during execution of com-
mand: resync
```

```
RMAN-07004: unhandled exception during command execution on chan-
nel default
```

```
RMAN-20001: target database not found in recovery catalog
```

**The target database is not registered in the catalog.**

- 3 Register target database in the recovery catalog at the RMAN prompt.

```
RMAN> register database;
```

- 4 Using RMAN, list all the database incarnations registered in the catalog.

```
RMAN> list incarnation of database;
```

```
RMAN-03022: compiling command: list
```

```
List of Database Incarnations
```

| DB Key | Inc Key | DB Name | DB ID      | CUR Reset SCN | Reset Time |
|--------|---------|---------|------------|---------------|------------|
| 101    | 102     | NEW8I   | 3579390644 | YES 425111    | 20-MAY-99  |

- 5 Enter the RESET DATABASE command at the RMAN prompt. What happens?

```
RMAN> reset database;
```

```
RMAN-03022: compiling command: reset
```

```
RMAN-03023: executing command: reset
```

```
RMAN-03026: error recovery releasing channel resources
```

```
RMAN-00571:=====
```

```
RMAN-00569:===== ERROR MESSAGE STACK FOLLOWS=====
```

```
RMAN-00571:=====
```

```
RMAN-03006: non-retryable error occurred during execution of command: reset
```

```
RMAN-07004: unhandled exception during command execution on channel default
```

```
RMAN-20009: database incarnation already registered
```

- 6 View the \$HOME/LABS/crebkup.sql script, and then run it as user system by logging in to target database using SQL\*PLUS to create an online operating system copy of the system tablespace data file in your \$HOME/BACKUP directory.

```
more $HOME/LABS/crebkup.sql
```

```
set head off
```

```
set feedback off;
```

```
spool $HOME/sysbkup.sh
```

```
select 'cp ' || name || ' $HOME/BACKUP/system01.cpy'
```

```
from v$datafile
```

```
where ts#=0;
```

```
spool off
```

```
ALTER TABLESPACE SYSTEM BEGIN BACKUP;
```

```
!chmod 775 $HOME/sysbkup.sh
```

```
!$HOME/sysbkup.sh
```

```
ALTER TABLESPACE SYSTEM END BACKUP;
```

```
set head on
```

```
set feedback on;
```

```
$ sqlplus system/manager
```

```
SQL> @$HOME/LABS/crebkup.sql
```

```
cp /u03/user/db01/DATA/DISK1/system01.dbf $HOME/BACKUP/system01.cpy
```

- 7 Using RMAN, add the backup made in step 6 to the catalog.

```

RMAN> catalog datafilecopy
 2> '$HOME/BACKUP/INC0/system01.cpy'
 3> tag 'SYSCPY0630';

RMAN-03022: compiling command: catalog
RMAN-03023: executing command: catalog
RMAN-08050: cataloged datafile copy
RMAN-08513: datafile copy filename=/u03/user/db01/BACKUP/
system01.cpy recid=
RMAN-03023: executing command: partial resync
RMAN-08003: starting partial resync of recovery catalog
RMAN-08005: partial resync complete

```

- 8 Using RMAN, confirm that the data file has been added to the recovery catalog.

```

RMAN> list copy;

RMAN-03022: compiling command: list

List of Datafile Copies
Key File S Completion time Ckp SCN Ckp time Name

178 1 A 21-JUL-99 35892 21-JUL-99 /u03/user/
db01/BACKUP/INC0/system01.cpy

```

- 9 Use the RMAN command to remove the backup of the system data file from the recovery catalog. *Do not* remove the file from the disk.

```

RMAN> change datafilecopy
 2> '/u03/user/db01/BACKUP/INC0/system01.cpy'
 3> uncatalog;

RMAN-03022: compiling command: change
RMAN-06119: uncataloged datafile copy
RMAN-08513: datafile copy filename=/u03/user/db01/BACKUP/INC0/
system01.cpy recid=1 stamp=371398288

```

- 10 Use the REPORT command to determine which data files have not yet been backed up by RMAN today.

```
RMAN> report need backup days 1;
```

```
RMAN-03022: compiling command: report
```

```
Report of files whose recovery needs more than 1 days of archived logs
```

```
File Days Name
```

```

1 2 /u03/user/db01/DATA/DISK1/system01.dbf
2 2 /u03/user/db01/DATA/DISK2/rbs01.dbf
3 2 /u03/user/db01/DATA/DISK3/data01.dbf
4 2 /u03/user/db01/DATA/DISK2/temp01.dbf
5 2 /u03/user/db01/DATA/DISK2/indx01.dbf
6 2 /u03/user/db01/DATA/DISK3/oemrep01.dbf
7 2 /u03/user/db01/DATA/DISK1/query01.dbf ;
```

- 11 Connect to your recovery catalog database using SQL\*Plus, and query RC\_DATAFILE\_COPY to confirm that the data file has been removed from the recovery catalog.

```
$ sqlplus db01/db01@db16;
```

```
SQL*Plus: Release 8.1.5.0.0 - Production on Wed Jul 21 14:32:38
1999
```

```
(c) Copyright 1998 Oracle Corporation. All rights reserved.
```

```
Connected to:
```

```
Oracle8i Enterprise Edition Release 8.1.5.0.0 - Production
```

```
With the Partitioning and Java options
```

```
PL/SQL Release 8.1.5.0.0 - Production
```

```
SQL> select name, db_name, file# from rc_datafile_copy;
```

```
NAME
```

```

DB_NAME FILE#

/u03/user/db01/BACKUP/INC0/system01.cpy
DB01 1
```

**12** Create a script to make a whole database backup with following information:

Name of script:nightback  
 Channel name: dbnD (n is the student account number)  
 Channel type Disk  
 Format \$HOME/BACKUP/%b%d%s%p  
 Level Database (No archive logs)  
 tag nback  
 DO NOT RUN THIS SCRIPT NOW.

```

RMAN> create script nightback {
 2> allocate channel db01D type disk;
 3> backup format '$HOME/BACKUP/%d%s%p'
 4> (database);
 5> release channel db01D;
 6> }

```

RMAN-03022: compiling command: create script

RMAN-03023: executing command: create script

RMAN-08085: created script nightback

**13** Query the recovery catalog and verify whether the script has been created.

```

$ sqlplus db01/db01@db16
SQL> select * from rc_stored_script;

```

| DB_KEY | DB_NAME | SCRIPT_NAME |
|--------|---------|-------------|
| 1      | DB01    | nightback   |

## Practice 12-1 Solutions

### Backup Using RMAN

- 1 What are the two supported backup types for Recovery Manager? List some of the differences between the two backup types.

**Two types of backup supported by the recovery manager are:**

**Backupset and Copy.**

**Backup set comprises of backup of one or more database files, while the copy contains a backup of only one datafile.**

**Copy can be made to a disk only while backup set can be taken to disk or tape.**

**Copy contains all the blocks of the input file (even the unused blocks in datafiles) while backup set may contain only the used blocks.**

**Copies operate on single files at file level while backup sets operate on files and their logical groups (such as Tablespace, Database).**

- 2 Create an image copy of data files belonging to SYSTEM tablespace. The copy should be placed in the \$HOME/BACKUP/INC0 directory with the name of sys0101.cpy. The tag should be SYSTEM01.

```
RMAN> run {
 allocate channel db01d type disk;
 copy datafile '$HOME/DATA/DISK1/system01.dbf' to
 '$HOME/BACKUP/INC0/sys0101.cpy'
 tag 'SYSTEM01';
}
```

- 3 Create a script to back up the database following these guidelines:
  - The name of the script should be FULLBACK.
  - Make a backup set.
  - The incremental level should be 0.
  - Backup pieces should be on disk in the \$HOME/INC0 directory.
  - The format string should include FULL%n\_%s.%p.
  - Do not include archive logs.
  - Place four files per backup set.
  - Parallel the backup operation in three sessions.



- ```

RMAN> create script FULLBACK {
allocate channel db01D1 type disk;
allocate channel db01D2 type disk;
allocate channel db01D3 type disk;
backup incremental level 0 filesperset 4
format '$HOME/BACKUP/INC0/FULL%n_%s_%p'
(database include current controlfile);
release channel db01D1;
release channel db01D2;
release channel db01D3;
}

```
- 4 Run the script created in step 3 and verify completion of the backup.

```

RMAN> run {execute script FULLBACK;}
RMAN> list backup;

```
 - 5 Using RMAN, back up archive logs generated today to \$HOME/BACKUP/INC1 directory.

```

RMAN> run {
allocate channel db01D1 type disk format '$HOME/BACKUP/INC1/%U';
backup archivelog from time 'SYSDATE-1';
release channel db01D1;
}

```
 - 6 Log in to target database using SQL*PLUS as user `scott`, and run the script `moreemp.sql`. Create a backup of DATA01 tablespace with following guidelines:
 - It should be a backup set.
 - The incremental level should be 2.
 - Do not include control file.
 - Backup piece should be on disk in the \$HOME/INC2 directory.

```

sqlplus scott/tiger
SQL> @moreemp.sql
SQL> exit
$ rman target / catalog db01/db01@db16
RMAN> run {
allocate channel db01D1 type disk;
backup incremental level 2
format '$HOME/BACKUP/INC2/data01.lv2'
(tablespace DATA01);
release channel db01D1;
}

```
 - 7 Obtain a listing of all data files that have not been backed up in the last two days.

```

RMAN> report need backup days 2;

```

Practice 13-1 Solutions

Restore and Recover Using RMAN

It is assumed that the previous practice in Lesson 12 has been completed successfully.

- 1 Obtain a list of backup sets registered in the catalog.

```
RMAN> list backupset;
```

- 2 Obtain a list of copies listed in the catalog.

```
RMAN> list copy;
```

- 3 Run the script `breakdb.sql` as `sysdba` user in a `SQL*PLUS` session. Then, using RMAN, recover the `DATA01` tablespace.

- 4 You have determined that `DISK3` (`$HOME/DATA/DISK3`) is corrupted. You must relocate all the files on `DISK3` to another location. You determined that `DISK4` has sufficient space. Using RMAN, relocate all the data files in `DISK3` to `DISK4`.

```
RMAN> startup mount pfile=$HOME/LABS/initU1.ora;
run{
  allocate channel db01D1 type disk;
  set newname for datafile 3 to '$HOME/DATA/DISK4/data01.dbf';
  set newname for datafile 6 to '$HOME/DATA/DISK4/oemrep01.dbf';
  restore database;
  switch datafile all;
  recover database;
  sql 'alter database open';
}
```

C

Hints

Practice 2-1 Hints

- 1 What is the V\$ view that you must query for the names of all data files in the database?
Hint: V\$DATAFILE.
- 2 What are the V\$ views that you must query for finding the current online redo log and names of all redo logs in the database?
Hint: V\$LOG and V\$LOGFILE.
- 3 What is the V\$ view that you must query for the names of all control files in the database?
Hint: V\$CONTROLFILE.
- 4 Name the V\$ view that you should check to find the name of the database before dropping tables or shutting down the database.
Hint: V\$DATABASE.
- 5 Name the V\$ view that you must query to locate processes still connected to the instance before shutting down the database.
Hint: V\$PROCESS.
- 6 Describe the significance of the parameters LOG_CHECKPOINT_INTERVAL and FAST_START_IO_TARGET in instance recovery.
Hint: When set, the LOG_CHECKPOINT_INTERVAL ensures that the target position of the checkpoint (in the redo log) does not lag behind the tail by more than the number of blocks specified.
The dynamic initialization parameter FAST_START_IO_TARGET allows to limit the number of blocks that must be read for crash or instance recovery.
- 7 What `init.ora` parameter configures the memory area in the SGA which buffers recovery information in memory before being written to disk?
Hint: LOG_BUFFER parameter.
- 8 What is the large pool, when is it used, and what `init.ora` parameter configures it?
Hint: The large pool is a separate memory area in the SGA and is configured with the LARGE_POOL_SIZE parameter.

9 Optional: Set up mirroring of control files so as to have three control files.

Hint: To add a new control file or change the number or location of the control file, use the following steps:

- Shut down the database.
- Make a copy of the existing control file to a different device using operating system commands.
- Edit or add the `CONTROL_FILES` parameter and specify names for all the control files.
- Start up the database.

10 Optional: Set up mirroring of redo log files so as to have three members per group.

Hint: Use the SQL command `ALTER DATABASE ADD LOGFILE MEMBER`. Ensure that the redo members are on different disks.

Practice 3-1 Hints

Oracle Backup and Recovery Configuration

- 1 Connect to SQL*Plus as `sysdba` and place the database in mount state.
Hint: If the database is open, shut down and then issue the startup command with mount option.
- 2 List the parameters `LOG_ARCHIVE_DEST`, `LOG_ARCHIVE_START`, and `LOG_ARCHIVE_FORMAT`, and note the values.
Hint: From within SQL*Plus, use the `SHOW` command, or query the view `V$PARAMETER`.
- 3 Execute the command `ARCHIVE LOG LIST`. Note the log mode of the database and whether automatic archival is enabled.
- 4 Set the database in archivelog mode.
Hint: In the SQL*PLUS session, use the command `ALTER DATABASE ARCHIVELOG`.
- 5 Open the database.
- 6 Shut down the instance with `IMMEDIATE` option.
- 7 Edit the `init.ora` file to:
 - Enable archiving
 - Archive log files to two destinations: `$HOME/ARCHIVE` and `$HOME/ARCHIVE2` (The `$HOME/ARCHIVE` is mandatory, and `$HOME/ARCHIVE2` optional.)
 - Use the archiving format of `arch_%s.arc`
 - Spawn two archive processes at instance start**Hint:** As indicated, edit the `init.ora` file and include or correct the appropriate parameters.
- 8 While editing the `init.ora` file, uncomment the `ROLLBACK_SEGMENTS` parameter to place the rollback segments `RBS01` and `RBS02` when the database starts.
Hint: Edit the parameter `ROLLBACK_SEGMENTS` in `init.ora` file.
- 9 Start up and open the database.
- 10 Verify that two archive processes are running.
Hint: From the OS, use the `PS -EF` command.

- 11** Execute the `ALTER SYSTEM SWITCH LOGFILE` command twice, then list the values of the `ARCHIVE` parameters. Do you see any archived log files? What is the format of the filename?

Hint: Use the `SHOW` command in `SQL*PLUS` session. Then use the “`ls`” command at the OS.

Practice 4-1 Hints

- 1 While the database is open, connect to the database as `sys` or `system` and using `V$` and Data Dictionary Views, make a list of all of the files that must be backed up for a whole offline database backup.

Hint: Query the `V$CONTROLFILE`, `V$LOGFILE` and `V$DATAFILE` views.

- 2 Shut down the database in `IMMEDIATE` mode. Make two whole offline database backups using the operating system commands. Place one in the `$HOME/DONTOUCH` directory, and the other in `$HOME/BACKUP`.

Hint

```
$ cp -r $HOME/DATA/* $HOME/DONTOUCH
$ cp $ORACLE_HOME/dbs/orapwU1 $HOME/DONTOUCH
$ cp -r $HOME/DATA/* $HOME/BACKUP
$ cp $ORACLE_HOME/dbs/orapwU1 $HOME/BACKUP
```

- 3 Start the instance.
- 4 Connect as `scott/tiger` and execute the `$HOME/LABS/deptdata.sql` and `$HOME/LABS/empdata.sql` scripts to create transactions against the database.
- 5 Connect as `system/manager` and take an open backup of the `DATA01` tablespace. Copy the file to `$HOME/BACKUP` directory. Make sure that you do not overwrite another copy.

Hint

- Use the `ALTER TABLESPACE DATA01 BEGIN BACKUP` command.
 - Use an operating system command to copy the data file associated with the `DATA01` tablespace.
 - Use the `ALTER TABLESPACE DATA01 END BACKUP` command.
- 6 Use the `ALTER DATABASE` command to back up the control file to trace. Connect to `SQL*Plus` and execute the `$HOME/LABS/spid.sql` script to identify the trace file. Host to the operating system, display the files in the directory and copy the trace file to `$HOME/BACKUP/cntrl.sql`. Using an editor, remove the comments from the trace file.
 - 7 Create a binary copy of the control file and put it in the `$HOME/BACKUP` directory. Name the backup copy `cntrl1.bkp`.

Hint: Use the `ALTER DATABASE BACKUP CONTROLFILE TO` command.

Practice 5-1 Hints

Complete Database Recovery: Noarchivelog Mode

- 1 Shutdown the database and disable automatic archival (ARC0) by editing the `init.ora` parameter file, then start and mount the database. Set the database in noarchivelog mode, and then open the database. Ensure the desired status by issuing ARCHIVE LOG LIST command.
Hint: Shutdown the database. Then mount the database.
 Use ARCHIVE LOG LIST to note that the database is in NOARCHIVELOG mode.
- 2 Shutdown the database and perform a full, closed backup by using the operating system command. Copy the database files (control files, redo log files and data files to the `$HOME/BACKUP/NOARC` directory only.) Start the instance and open the database.
- 3 Start SQL*Plus and connect by using `scott/tiger`, and run the `$HOME/LABS/newemp.sql` script. This script creates new users and tables adding important information to it.

```
SQL> @$HOME/LABS/newemp.sql
```
- 4 Connect as `system/manager` and run the following script to record the name of data files that contain the table NEWEMP:
Hint: Run a query that joins `dba_tables` and `dba_data_files` to obtain the filename.
- 5 Run the `$HOME/LABS/breakdb.sql` as `sys` or `system` in SQL*Plus to simulate failure.
- 6 Attempt to restart the database normally. What happened?
Hint: Use the STARTUP OPEN command.
 The database cannot locate the files for the DATA01 tablespace because of perceived media failure.
- 7 Shut down the database and use the appropriate operating system command to replace the current database with latest backup (from NOARC directory to DATA directory).
Hint: Shut down the database. Use the COPY (cp on UNIX) command.
- 8 Start up and open the database so that it will be available to all users.
- 9 Connect to the database as `scott/tiger` to execute a query against the NEWEMP table. What happened and why?
Hint: The table does not exist because it was created after the previous backup.
- 10 What conclusions can you make about offline backups and recovery for databases in noarchivelog mode?
Hint: In noarchivelog mode, you may not be able to recover data entered after the latest backup.

Practice 5-2 Hints

Complete Database Recovery: Archivelog Mode

- 1 Query the V\$DATABASE view to see what mode the database is in. Use ARCHIVE LOG LIST to check the status of archivelog mode and automatic archiving.
Hint: Execute ARCHIVE LOG LIST.
- 2 Shutdown the instance and uncomment the LOG_ARCHIVE_START parameter in the init.ora file.
Mount the database to enable archiving.
Hint: Mount the database. Then use the ALTER DATABASE ARCHIVELOG command.
- 3 Check to make sure archiving is successful by using the ARCHIVE LOG LIST command. Note the current log sequence number.
- 4 Perform a closed database backup. Store the backup in the \$HOME/BACKUP directory.
Hint: Shut down the database and use an operating system command such as CP -R in UNIX or XCOPY in NT.
- 5 Start SQL*Plus and connect by using scott/tiger and run the \$HOME/LABS/newemp.sql script. This script simulates the creation of a new table and users adding important information to it.
- 6 Connect as system/manager and run the script \$HOME/LABS/checktbs.sql.
Hint: Note the name of the data file that the script returns.
- 7 Run the \$HOME/LABS/breakdb.sql script to simulate hardware failure.
- 8 Attempt to restart the database normally. What happened?
Hint: The database is left in MOUNT state. One data file is not available.
- 9 The database cannot locate the files for the DATA01 tablespace because of perceived media failure. Since archiving is enabled, you can now perform a complete recovery.
Restore the data files for the DATA01 tablespace from the backup that you made in step 4.
Hint: Use the CP -P command in UNIX. Use the COPY command in NT.
- 10 Use the RECOVER DATABASE command to recover the database.

- 11** When recovery is complete, open the database to make it available for all users.
Hint: Use the ALTER DATABASE OPEN command.
- 12** Query the DBA_TABLESPACES view to see if DATA01 is online. Make sure the tablespace is online.
Hint: Query the TABLESPACE_NAME and STATUS columns of the view DBA_TABLESPACES.
- 13** Connect to the database as `scott/tiger` and execute a query against the NEWEMP table. What happened and why?
Hint: The table NEWEMP has been recovered. It is possible to recover to a current time when the database is in ARCHIVELOG mode.
- 14** Connect as `system/manager` and query the V\$LOG view and note the sequence number. Compare this value with the value in step 3. What conclusions can you make about complete recovery?
Hint: Query V\$LOG for STATUS = 'CURRENT' and note the SEQUENCE# column.

Practice 6-1 Hints

Recovering from User Failure: Incomplete Recovery

- 1 If you are unsure whether you have a valid backup from the previous exercise, then perform either a whole closed or opened database backup. Store the backup in the \$HOME/BACKUP directory.
Hint: Use the CP -RP command in UNIX or XCOPY /S on Windows NT.
- 2 Start SQL*Plus and connect by using `scott/tiger` and insert rows into the NEWEMP table by executing the following statement:

```
SQL> insert into newemp select * from newemp;
```
- 3 Select a count of the rows in the NEWEMP table. Note the number of rows.
Hint: Use the `SELECT COUNT(*)` command.
- 4 Connect as `system/manager` and run the following commands:

```
SQL> select f.file_name from dba_tables t, dba_data_files f
      2 where table_name = 'NEWEMP' and
      3 t.tablespace_name=f.tablespace_name;
```

Record the filename of all data files for the tablespace.
- 5 Record the current system time using an operating system command.
Hint: Use the `DATE` command on UNIX or `TIME /T` command on Windows NT.
- 6 Record the current online log sequence number by querying `V$LOG`.
Hint: Query the column `SEQUENCE#` in the view `V$LOG`.
- 7 Connect using `scott/tiger` and add rows to the NEWEMP table by executing the following command:

```
SQL> insert into newemp select * from newemp;
```
- 8 Get a count of the rows in the NEWEMP table.
Hint: Use the `select COUNT(*)` command.
- 9 Run the `$HOME/LABS/breaktab.sql` script to simulate a user accidentally dropping the NEWEMP table.
- 10 Attempt to query the NEWEMP table. What happened?
Hint: The table no longer exists.
- 11 The database cannot locate the NEWEMP table. Since automatic archiving is enabled, you have all the archives, and you know the approximate time of failure, you can now perform an incomplete recovery to bring back the table.
Shut down and then mount the database.

- 12 Restore all data files from the backup that you made in step 1.
Note: Check the V\$DATAFILE view with the `$HOME/LABS/cpdbfile.sh` script. If the files and locations are identical, then run this script. Otherwise, you may modify this file to match your database structure.
- 13 Recover the database until the time you noted in step 4.
Hint: Use the `RECOVER DATABASE UNTIL TIME` command.
- 14 When recovery is complete, open the database using the `RESETLOGS` option to enable access for all users.
Hint: Use the `ALTER DATABASE OPEN RESETLOGS` command.
- 15 Connect to the database as `scott/tiger` and execute a query against the `NEWEMP` table. What happened and why?
Hint: The table has been recovered.
- 16 Connect as `system/manager` and query the `V$LOG` view and note the sequence number. Compare this value with the value in step 5. What conclusions can you make about incomplete recovery?
- 17 Take a whole offline backup. Store the backup in the `$HOME/BACKUP` directory.

Practice 7-1 Hints

Use the Export and Import utilities to save flat file copies of data offline.

- 1 Invoke the Export utility to export the EMP and DEPT tables in the `scott` schema.

Hint: From the operating system prompt, execute the EXP command.

- 2 With SQL*Plus, connect as `scott`. Drop the EMP and DEPT tables.
- 3 Restore the EMP and DEPT tables by using the import utility.

Hint

- From the operating system prompt, execute the IMP command.
- Query the EMP and DEPT tables.

Practice 8-1 Hints

Starting the Oracle Database with a Missing Data File

- 1 Connect as sysdba.
- 2 Shut down the database by using IMMEDIATE.
- 3 From the operating system prompt, rename the data file `$HOME/DATA/DISKS/temp01.dbf` to `$HOME/DATA/DISKS/temp01.old`.
- 4 Connect as sysdba and start the database. What error message do you receive?
Hint: You notice a data file is missing.
- 5 Drop the data files for the TEMP tablespace.
Hint: Use the OFFLINE DROP clause of the ALTER DATABASE command.
- 6 Open the database.
- 7 Drop the TEMP tablespace.
Hint: You must use INCLUDING CONTENTS option.
- 8 Create the TEMP tablespace using the filename `temp01.dbf` with a size of 50K.
Hint: Note the directory in which you have created the file.

Reconstruct Lost Control Files

- 1 Connect as sysdba.
- 2 Create a trace file to keep the syntax for recreating the control files in case of loss.
Hint: Use the ALTER DATABASE BACKUP CONTROLFILE TO TRACE command.
- 3 Shut down the database by using IMMEDIATE.
- 4 From the operating system prompt, remove the control files `$HOME/DATA/DISK1/cntrl_1.ctl` and `$HOME/DATA/DISK2/cntrl_2.ctl`.
- 5 Connect as sysdba and start the database. What error message do you receive?
Hint: The control file cannot be found in the specified place.
- 6 Execute the `$HOME/BACKUP/crecntl.sql` trace file script to re-create the control files.
Hint: Ensure that the comment lines are marked as REM and not as #.
- 7 Shut down the database. Start up the database to ensure that your database is functioning correctly.

Recover a Read-Only Tablespace

- 1 Connect as sysdba.
- 2 Shut down the database using IMMEDIATE.
- 3 From the operating system prompt, rename the data file \$HOME/DATA/DISK1/query01.dbf to \$HOME/DATA/DISK1/query01.old.
Hint: On UNIX use the MV command. On NT use REN command.
- 4 Connect as sysdba and start the database. What error message do you receive?
Hint: One of the data files is missing.
- 5 Restore the data files for the QUERY_DATA tablespace.
Hint: Copy from the backup stored in the \$HOME/BACKUP directory.
- 6 Open the database.

Practice 9-1 Hints

- 1 Execute the ALTER SYSTEM SWITCH LOGFILE command twice. Change to the BDUMP directory. View the alert_<SID>.log file.

Hint: You notice that the occurrence of Log Switch is recorded in alert log.

- 2 Connect to SQL*Plus and use the SHOW command to check the values of the parameter DB_BLOCK_CHECKSUM.
- 3 Run the DBVERIFY utility against the user01.dbf data file. Do you see any corrupted blocks?

Hint: Use the BLOCKSIZE parameter.

Use of LogMiner Utility

- 1 A new employee Scott inserts a new row into SCOTT.EMP table.

Hint: Note the structure of EMP table before issuing insert command.

- 2 The next day, the DBA is asked to cancel the wrong insertion made some time the previous day by Scott on SCOTT.EMP table. The user, however, cannot remember which row was inserted.

Initialize the LogMiner utility to undo the SQL insert statement executed by Scott on SCOTT.EMP table the previous day.

Hint: Set the value for the UTL_FILE_DIR parameter. Build the LogMiner dictionary file using the BUILD procedure of DBMS_LOGMNR_D package.

- 3 Specify redo log files to be analyzed.

Hint: Use the ADD_LOGFILE procedure of DBMS_LOGMNR package, first with NEW and then with ADDFILE options.

- 4 Start LogMiner analysis.

Hint: Use the START_LOGMNR procedure of DBMS_LOGMNR package.

- 5 Find the insertion made the day before (in reality, during the last five minutes).

Hint: Query V\$LOGMNR_CONTENTS view. Check the columns SQL_REDO and SQL_UNDO.

- 6 Cancel the insertion.

Hint: Run the statement in SQL_UNDO from a SQL session.

- 7 Stop LogMiner analysis.

Hint: Use the END_LOGMNR procedure of DBMS_LOGMNR package.

Practice 11-1 Hints

- 1 Connect to your target database and recovery catalog using RMAN.
Hint: Use target and catalog options in RMAN.
- 2 Execute the command to resync the recovery catalog with control file of the target database. What happened? Why?
Hint: Target database has not yet been registered with the catalog.
- 3 Register the target database with the recovery catalog.
- 4 Using RMAN, list all the database incarnations registered in the catalog.
Hint: Use the appropriate LIST command.
- 5 Enter the RESET DATABASE command. What happens? Why?
Hint: RMAN reports that reset would not be necessary.
- 6 View the script `$HOME/LABS/crebkup.sql` and then run it as user `system` by logging in to the target database using SQL*PLUS to create an online operating system copy of the data files of SYSTEM tablespace in your `$HOME/BACKUP` directory.
Hint: Use the MORE command to view the file in UNIX. Use the TYPE command in Windows NT. Run the script in SQL*PLUS as user `system`.
- 7 Using RMAN, add the backup made in step 6 to the catalog.
Hint: Use the CATALOG command.
- 8 Using RMAN, confirm that the data file has been added to the recovery catalog.
Hint: Use the list COPY command.
- 9 Use the RMAN command to remove the backup of the system data file from the recovery catalog. *Do not* remove the file from the disk.
Hint: Use the CHANGE ... UNCOPY command.
- 10 Use the REPORT command to determine which data files have not yet been backed up by RMAN today.
Hint: Use the REPORT command.
- 11 Connect to your recovery catalog database using SQL*Plus, and query RC_DATAFILE_COPY to confirm the data file has been removed from the recovery catalog.
Hint: Connect to the recovery catalog as the user with RECOVERY_CATALOG_OWNER privilege, and query the RC_DATAFILE_COPY view.

12 Create a script to make a whole database backup with following information:

```
Name of script:nightback
Channel name:   dbnD (n is the student account number)
Channel type    Disk
Format          $HOME/BACKUP/%b%d%s%p
Level           Database (No archive logs)
tag             nback
DO NOT RUN THIS SCRIPT NOW.
```

Hint: Use the CREATE SCRIPT command.

13 Query the recovery catalog and verify whether the script has been created.

Hint: Connect to the recovery catalog as the user with RECOVERY_CATALOG_OWNER privilege, and query the RC_STORED_SCRIPT view.

Practice 12-1 Hints

Backup Using RMAN

- 1 What are the two supported backup types for Recovery Manager? List some of the differences between the two backup types.
- 2 Create an image copy of data files belonging to SYSTEM tablespace. The copy should be placed in \$HOME/BACKUP/INC0 directory with the name of sys0101.cpy. The tag should be SYSTEM01.
- 3 Create a script to back up the database following these guidelines:
 - The name of the script should be FULLBACK.
 - Make a backup set.
 - The incremental level should be 0.
 - Backup pieces should be on disk in the \$HOME/INC0 directory.
 - The format string to include FULL%n_%s.%p.
 - Do not include archive logs.
 - Place four files per backup set.
 - Parallel the backup operation in three sessions.
- 4 Run the script created in step 3 and verify completion of the backup.
- 5 Using RMAN, back up archive logs generated today to \$HOME/BACKUP/INC1 directory.
Hint: Use the FROM TIME 'SYSDATE-1' option.
- 6 Log in to the target database using SQL*PLUS as user scott, and run the script moreemp.sql. Create a backup of DATA01 tablespace with the following guidelines:
 - It should be a backup set.
 - The incremental level should be 2.
 - Do not include the control file.
 - Backup piece should be on disk in the \$HOME/INC2 directory.
- 7 Obtain a listing of all data files that have not been backed up in the last two days.

Practice 13-1 Hints

Restore and Recover Using RMAN

It is assumed that the previous practice in Lesson 12 has been completed successfully.

- 1 Obtain a list of backup sets registered in the catalog.
- 2 Obtain a list of copies listed in the catalog.
- 3 Run the script `breakdb.sql` as `sysdba` user in a `SQL*PLUS` session. Then, using RMAN, recover the `DATA01` tablespace.

Hint: Use the `RECOVER TABLESPACE` option if database is open.

- 4 You have determined that `DISK3` (`$HOME/DATA/DISK3`) is corrupted. You must relocate all the files on `DISK3` to another location. You determined that `DISK4` has sufficient space. Using RMAN, relocate all the data files in `DISK3` to `DISK4`.

Hint: Use the `SWITCH DATA FILE` option for relocating files.

D

Workshop Scenarios

Workshop Scenarios

Scenario 1: Loss of INACTIVE Online Redo Log Group

Scenario 2: Loss of CURRENT Online Redo Log Group

Scenario 3: Loss of Control Files

Scenario 4: Loss of Media

Scenario 5: Loss of an Online Rollback Segment Data File (Open or Closed Database)

Scenario 6: Loss of a System Tablespace Data File

Scenario 7: Loss of a Non-System, Non-Rollback Segment Data File

Scenario 8: Recover from User Errors

Scenario 9: Failure During Hot Backup

Scenario 10: Configuring Recovery Catalog

Scenario 11: Missing Data File

Scenario 12: Loss of a Data File and Missing Archive Log File

Scenario 13: Recover a Lost Data File with No Backup

Scenario 14: Missing Mirrored Online Redo Log Files

Scenario 15: Loss of a Control File and Read-Only Tablespace

Scenario 16: Loss of Non-Essential Data File When Database Is Down

Note: Edit `init.ora` and comment out `REMOTE_LOGIN_PASSWORD_FILE` entry.

Grant DBA to scott.

Ensure that the DATABASE is in ARCHIVELOG mode.

Timings for scenarios may vary, but you can approximate 30 minutes per scenario.

To ensure uniformity, you can copy the files from DONTouch directory to DATA directory (`cp -r $HOME/DONTouch $HOME/DATA`).

Edit the `wksh` script in `$HOME/LABS` directory to match the database names. It is presumed in the script that database names for students will be U1-U15.

Make a control file backup to trace for each database. Edit it and retain it in DONTouch and BACKUP directories.

After each scenario is complete, shut down the database. Copy the data files, control files and the redo log files from DONTouch directory. You may want to remove the archive logs at the end of each scenario.

Scenario 1: Loss of INACTIVE Online Redo Log Group

Solution Outline

- 1 Shut down the instance.
- 2 MOUNT the instance.
- 3 If the lost log file was archived, check the V\$LOG view.
- 4 If there is an offline data file, check V\$DATAFILE that requires the cleared unarchived log to bring it online. Issue the ALTER DATABASE CLEAR LOGFILE command; the keywords UNRECOVERABLE DATAFILE are required. The data file and its entire tablespace must be dropped from the database because the redo necessary to bring it online is being cleared, and there is no copy of it.
- 5 Add a new redo log group by using the information noted on the Database Configuration Checklist, such as: “Add a redo log file member using the information on the Database Configuration Checklist.”

```
ALTER DATABASE ADD LOGFILE GROUP 3  
    '$HOME/DATA/DISK3/redo0301.log' SIZE 150K
```

- 6 Drop the damaged redo log file group:

```
SQL>ALTER DATABASE DROP LOGFILE GROUP X ; Appropriate Group No.
```

- 7 Add a member to the Logfile group 3:

```
ALTER DATABASE ADD LOGFILE MEMBER  
    '$HOME/DATA/DISK4/redo0302.log' TO GROUP 3;
```

- 8 Determine if a full offline backup is required, and perform one if necessary.
- 9 Ensure that the instance is started and that the database is open.
- 10 Run the `moreemp.sql` script.

For more information, refer to the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 2: Loss of CURRENT Online Redo Log Group

Solution Outline

- 1 Start the instance if necessary.
- 2 Attempt to alter the database and drop the redo log group. You will receive an error stating that you cannot drop the current redo log.
- 3 Shut down the instance.
- 4 Review the trace files if any and the `alert.log` file.
- 5 Copy the data files and redo log files from the backup directory into their respective `DISKn` directories.
- 6 MOUNT the instance.
- 7 Query the `V$LOG` view to determine the `sequence#` of the CURRENT log group.
- 8 Run ARCHIVE LOG LIST.
- 9 Recover the database until it is canceled. Cancel when the current log is suggested.
- 10 When recovery is complete, open the database with the RESETLOGS command.
- 11 View the `alert.log` file for the recovery that was applied.
- 12 Determine if a full offline backup is required and perform one if necessary.
Remove the archive log files from the archive directory.
- 13 Remove the `alert.log` and trace files from the trace directory.
- 14 Ensure that the instance is started and the database is open.
- 15 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 3: Loss of Control Files

Solution Outline

- 1 Start the instance if necessary.
- 2 Shut down the instance if the start failed.
- 3 Connect to SQLPLUS and start the instance in NOMOUNT mode.
- 4 Run the trace file script to recreate the control file.
- 5 Determine if a full offline backup is required and perform one if necessary.
- 6 Ensure that the instance is started and the database is open.
- 7 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 4: Loss of Media

Solution Outline

- 1 Mount the database. If one of the lost data files belongs to a rollback segment tablespace, shut down the instance comment out the `ROLLBACK_SEGMENTS` parameter in the `init.ora` file before placing instance in MOUNT state.
- 2 Determine which files to recover using `V$RECOVER_FILE` and `V$DATAFILE`.
- 3 Use the `ALTER DATABASE DATAFILE OFFLINE` command to take the datafiles offline so you can open the database.
- 4 Once the database is open, use the command `ALTER TABLESPACE <tablespace_name> OFFLINE IMMEDIATE` Restore missing files from the backup directory to another available device (`DISKn` directory).
- 5 Rename the files so the changes are recorded in the control file.
- 6 Issue the command `RECOVER DATAFILE` to recover each individual data file, or if all files of a tablespace are involved, then use the command:

```
'RECOVER TABLESPACE <tablespace_name>' to recover all datafiles for a selected tablespace.
```
- 7 When the files have been recovered, bring the tablespace(s) online.
- 8 Query the `V$DATAFILE` view to check the status of the files.
- 9 Query the `V$RECOVER_FILE` view to check the status of damaged files.
- 10 From the `$HOME` directory, create the subdirectory that was removed, such as, `mkdir $HOME/DATA/DISK3`. Also remember to set the correct privileges for Oracle to write to that directory by issuing `chmod 775 $HOME/DATA/DISK3`.
- 11 Take the tablespaces offline and make a physical copy of the data files to their original location, from the directory in step 4 to the directory in step 12.
- 12 Use the `ALTER DATABASE RENAME FILE` command to record the structural change in the control file.
- 13 Bring the tablespaces online.
- 14 Ensure that the instance is started and the database is open.
- 15 Determine if a full offline backup is required and perform one if necessary.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*
- Bulletin 1012943.6 in Appendix E

Scenario 5: Loss of File Containing Online Rollback Segment

Solution Outline

- 1 Start the instance. You will receive an error. (You can also review the trace file for indications of a corrupt rollback segment).
- 2 Reference the attached bulletin (1013221.6) for resolving rollback segment data file recovery.
- 3 Determine if a full offline backup is required and perform one if necessary.
- 4 Ensure that the instance is started and the database is open.
- 5 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*
- Bulletins 1013221.6 and 1010700.6 in Appendix E

Scenario 6: Loss of a Data File of System Tablespace

Solution Outline

- 1 Start the instance (it will fail).
- 2 Shut down the instance if the start failed.
- 3 Restore the data file (belonging to system tablespace) file from BACKUP DIRECTORY.
- 4 Use the MOUNT command to mount the instance.
- 5 Perform database recovery until you receive the message, "Media recovery complete."
- 6 Determine if a full offline backup is required and perform one if necessary.
- 7 Open the database.
- 8 Ensure that the instance is started and the database is open.
- 9 Run the `moreemp.sql` script as scott.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 7: Loss of a Non-System, Non-Rollback Segment Data File

Solution Outline

- 1 Start the instance if necessary.
- 2 Shut down the instance if the start failed.
- 3 Use the MOUNT command to mount the instance.
- 4 Open the database and put the related tablespaces offline using IMMEDIATE option.
- 5 Restore the data file from the BACKUP directory.
- 6 Perform recovery of the data file.
- 7 Bring the related tablespace online.
- 8 Run the `moreemp.sql` script.
- 9 Perform a full offline backup.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 8: Recover from User Errors

Solution Outline

- 1 Three recovery scenarios pertain to this failure. Remember, however, that the objective is to minimize downtime and data loss when determining which recovery method to use.
 - a You may restore the entire database using a point-in-time recovery, which means you will lose any transactions that occurred after the time to which you recover.
 - b You may restore the database at another location, export the table, then import the individual table into the primary database.
 - c Restore the table from an export file.
- 2 Ensure that the instance is started and the database is open.
- 3 Run the `moreemp.sql` script.
- 4 Determine if a full offline backup is required and perform one if necessary.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 9: Failure During Hot Backup

Solution Outline

- 1 Mount the database.
- 2 Query the view V\$RECOVER_FILE.
- 3 Query the view V\$BACKUP.
- 4 Determine which files were in backup mode when the database crashed.
- 5 Take the suspect file out of backup mode

```
ALTER DATABASE DATAFILE 'file_name' END BACKUP;
```
- 6 Open the database.
- 7 Query the V\$RECOVER_FILE, V\$BACKUP, and V\$DATAFILE views.
- 8 Determine if a full offline backup is required and perform one if necessary.
- 9 Ensure that the instance is started and the database is open.
- 10 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 10: Configuring a Recovery Catalog

Solution Outline

- 1 Connect to the catalog database and create a tablespace (with at least 30M) for your recovery catalog. (In the current case, you will create a catalog in the same database as your target.)

```
SQL> CREATE TABLESPACE RCAT_TS DATAFILE
      '$HOME/DATA/DISK4/rcat01.dbf' size 30M;
```

- 2 Create a recovery manager user in the database.

```
SQL> CREATE USER RMAN_DB1 IDENTIFIED BY RMAN
      2  DEFAULT TABLESPACE RCAT_TS
      3  TEMPORARY TABLESPACE TEMP
      4  QUOTA UNLIMITED ON RCAT_TS;
```

- 3 Grant roles and privileges to rman user.

```
SQL> GRANT CONNECT, RESOURCE, RECOVERY_CATALOG_OWNER TO
      RMAN_DB1;
```

- 4 Connect by using RMAN to the catalog database. (Do not use the TARGET option).

```
$ rman CATALOG RMAN_DB1/RMAN
Recovery Manager: Release 8.1.5.0.0 - Production
RMAN-06008: connected to recovery catalog database
RMAN-06428: recovery catalog is not installed
RMAN>
```

- 5 Issue the CREATE CATALOG command. Use the appropriate tablespace.

```
RMAN> CREATE CATALOG TABLESPACE RCAT_TS;
RMAN-06431: RECOVERY CATALOG CREATED
RMAN>
```

- 6 Connect to the target database.

```
RMAN> CONNECT TARGET /
RMAN-06005: connected to target database: DB01 (DBID=1118660359)
RMAN>
```

7 Register the database:

```

RMAN> REGISTER DATABASE;
RMAN-03022: compiling command: register
RMAN-03023: executing command: register
RMAN-08006: database registered in recovery catalog
RMAN-03023: executing command: full resync
RMAN-08002: starting full resync of recovery catalog
RMAN-08004: full resync complete

```

```
RMAN>
```

8 Check that the database has been registered.

```

RMAN> LIST INCARNATION OF DATABASE;
RMAN-03022: compiling command: list
List of Database Incarnations
DB Key Inc Key  DB Name  DB ID          CUR Reset  SCN  Reset Time
-----
          1      2      DB01  1118660359      YES    1    19-JUL-99
RMAN> exit
$ sqlplus rman_db1/rman
SQL> SELECT * FROM RC_DATABASE;
DB_KEY DBINC_KEY  DBID          NAME  RESETLOGS_CHANGE#  RESETLOGS
-----
1      2          1118660359    DB01  1                  19-JUL-99

```

Scenario 11: Missing Data File with No Backup

Solution Outline

- 1 Perform recovery in accordance with the attached bulletin.
- 2 Determine if a full offline backup is required and perform one if necessary.
- 3 Ensure that the instance is started and the database is open.
- 4 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*
- Bulletin 1005254.6 in Appendix E

Scenario 12: Loss of a Data File and Missing Archive Log File

Solution Outline

- 1 Restore the datafile to the correct directory.
- 2 Use the MOUNT command to mount the instance.
- 3 Begin database recovery. You will discover that you are missing an archived log file.
- 4 Shut down the instance.
- 5 Determine which archive log file you are missing. Check V\$RECOVERY_LOG to get the archival information, and check the LOG_ARCHIVE_DEST_n directories.
- 6 Restore your data files from the \$HOME/BACKUP directory, except the control files.
- 7 Perform a CANCELED-BASED recovery, canceling the operation at the appropriate point.
- 8 Open the database using the RESETLOGS option.
- 9 Perform full offline backup and remove the old archive files.
- 10 Ensure that the instance is started and the database is open.
- 11 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 13: Loss of Non-Essential Data File When Database Is Down

Solution Outline

- 1 Start the instance if necessary.
- 2 Shut down the instance if the start failed.
- 3 Mount the database.
- 4 Open the database.
- 5 Query the V\$RECOVER_FILE view and note the filename.
- 6 Alter the database to take the file offline and drop it.
- 7 Open the database.
- 8 Drop the INDX01 tablespace.
- 9 Create the INDX01 tablespace using the same file name noted in step 5 by using a size of 500K, and the reuse option.
- 10 Run the `index.sql` script.
- 11 Determine if a full offline backup is required and perform one if necessary.
- 12 Ensure that the instance is started and that the database is open.
- 13 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 14: Recover a Lost Data File from Archive Logs

Solution Outline

- 1 Use the MOUNT command to mount the instance.
- 2 Query the V\$RECOVER_FILE view.
- 3 Query the V\$DATAFILE view.
- 4 Alter the database and create the new data file as a new filespec. The filename is \$HOME/DATA/DISK4/new01.dbf, and the size is 500K.
- 5 Recover the data file.
- 6 Open the database.
- 7 Query the V\$RECOVER_FILE view.
- 8 Query the V\$DATAFILE view.
- 9 Determine if a full offline backup is required and perform one if necessary.
- 10 Ensure that the instance is started and that the database is open.
- 11 Verify whether SCOTT.EMP1 exists.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 15: Missing Mirrored Online Redo Log Files

Solution Outline

- 1 Start the instance if necessary.
- 2 Review the `alert.log` file and trace files for abnormal conditions.
- 3 Query the `V$LOGFILE` view.
- 4 Switch the log files and then query the `V$LOGFILE` view. The problem is that the only the mirrored redo log files are corrupted but the database is still available.
- 5 Correct the problem by adding new log files using the naming conventions on the database configuration checklist.
- 6 Determine if a full offline backup is required and perform one if necessary.
- 7 Ensure that the instance is started and that the database is open.
- 8 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

Scenario 16: Loss of a Control File and Read-Only Tablespace

Solution Outline

- 1 Start the instance if necessary.
- 2 Shut down the instance if the start failed.
- 3 View the `alert.log` file.
- 4 Restore the missing datafile and one control file from backup.
- 5 Edit your `init.ora` file.
- 6 Mount the database.
- 7 Recover the database using the backup control file option. (You may need to apply one of the online redo logs if necessary.)
- 8 Open the database using the RESETLOGS option.
- 9 Put the QUERY_DATA tablespace into READ ONLY mode.
- 10 Determine if a full offline backup is required and perform one if necessary.
- 11 Ensure that the instance is started and that the database is open.
- 12 Run the `moreemp.sql` script.

For more information, see the following publications:

- *Oracle8i Server Administrator's Guide*
- *Oracle8i SQL Reference Manual*

E

Worldwide Support Bulletins

Oracle Corporate Support Problem Repository

Missing Data File

- 1 Prob# 1005254.6 CANNOT STARTUP THE DATABASE BECAUSE A DATAFILE WAS REMO
- 2 Soln# 2031159.6 WORKAROUNDS TO OPEN UP THE DATABASE WITH A REMOVED DATA
- 3 Prob# 1005254.6 CANNOT STARTUP THE DATABASE BECAUSE A DATAFILE WAS REMO

Summary Cannot start up the database because a data file was removed from the file directory

Problem Description If a file is physically removed from the operating system directory, you may either get errors while the database is up or while the DBA is trying to start up the database.

Problem Explanation The Oracle Server verifies the existence and consistency of each data file registered in the control file after the database is successfully mounted. If the file is bad or is being unintentionally removed, the following error may occur:

```
ORA-01157: cannot identify data file %n - file not found
ORA-01110: data file %n: '%s'
```

The following errors may also occur when trying to do a normal shutdown or when DBWR attempts to write to the file that is being removed:

```
ORA-01116: error in opening database file %n
ORA-01110: data file %n: '%s'
ORA-07368: sfofi: open error, unable to open database file.
```

These errors are followed by an operating system specific error (for instance, error number 2 in most UNIX platforms).

Diagnostics and References

```
* {1942.6,Y,100} ORA-1110 DATA FILE 2: '/.../RBSDSV1.DBF"
* {2170.6,Y,100} ORA-1116 ERROR IN OPENING DATABASE FILE 2
* {4303.6,Y,100} ORA-11157
* {6179.6,Y,100} LOST A DATAFILE
```

Soln# 2031159.6 WORKAROUNDS TO OPEN UP THE DATABASE WITH A REMOVED DATA

Solution ID: 2031159.6
For Problem: 1005254.6
Affected Platforms: Generic: not platform specific
Affected Products: Oracle7 Server
Affected Components: RDBMS V07.XX
Affected Oracle Vsn: V07.XX

Summary Workarounds to open up the database with a removed data file

Solution Description *Warning: This solution can only be applied if the removed data file does NOT belong to the system tablespace or to a rollback tablespace. If the file belongs to the system tablespace or to a rollback tablespace, please contact Oracle customer support.*

There are cases in which the DBA inadvertently removes a data file from the file directory, maybe with the incorrect assumption that by removing the file, any reference to it from the Oracle Server is also removed. It may also be due to the fact that an operating system error or hardware problem rendered the file unreadable or inaccessible.

If the file is inaccessible by the Oracle Server, the DBWR may force the data file to go offline, in which case you would get the following error when trying to access the data file by any means:

```
ORA-01135, 00000, "file %s accessed for DML/query is offline"  
// *Cause: Attempted to access a data file that is offline  
// *Action: Bring the data file back online
```

In either case, the easiest way is to drop the entire tablespace that contains the data file. The steps to be executed from within SQL*DBA are:

1 STARTUP MOUNT

2 For each deleted data file, issue the command

```
ALTER DATABASE DATAFILE 'full path of filename' OFFLINE [DROP];
```

Note: You must use the DROP option if the database is in NOARCHIVELOG mode, because you cannot recover this file if you apply incomplete media recovery on it via the command ALTER DATABASE OPEN RESETLOGS. See the *SQL Reference Manual* for details.

3 ALTER DATABASE OPEN;

4 DROP TABLESPACE <tablespace> INCLUDING CONTENTS [CASCADE CONSTRAINTS];

Data Block Corruption

1 Prob# 1010640.6 DATA BLOCK CORRUPTION BULLETIN

2 Soln# 2058665.6 POSSIBLE WORKAROUNDS FOR ORA-1578 - BULLETIN
108491.543

Prob# 1010640.6 DATA BLOCK CORRUPTION BULLETIN

Problem ID: 1010640.6
Affected Platforms: Generic: not platform specific
Affected Products: Oracle7 Server
Affected Components: RDBMS Generic
Affected Oracle Vsn: Generic

Summary Data block corruption bulletin

+==+

Problem Description When there is a corrupt data block in the database, one of the most common errors you might receive when you try to access that corrupted block is ORA-1578.

Other errors you might also receive to indicate a corruption are:

ora-600 [3339]

ora-600 [3398]

Problem Explanation

```
ORA-01578, 00000, "ORACLE data block corrupted (file # %s, block #
%s)"
// *Cause: The data block indicated was corrupted, mostly due to
software // errors.
// *Action: Try to restore the segment containing the block
indicated. This
// may involve dropping the segment and recreating it. If there
// is a trace file, report the errors in it to your ORACLE
// representative.
```

The bulletin 108491.543 by EPITT discusses how to resolve the ora-1578, although the concepts may also be applied to the other corruption errors as well.

+==+

Diagnostics and References

* {2003.6,Y,100} ORA-1578

* {6085.6,Y,100} CORRUPTED DATABASE BLOCKS

Soln# 2058665.6 POSSIBLE WORKAROUNDS FOR ORA-1578 - BULLETIN
108491.543

Solution ID: 2058665.6
For Problem: 1010640.6
Affected Platforms: Generic: not platform specific
Affected Products: Oracle7 Server
Affected Components: RDBMS Generic
Affected Oracle Vsn: Generic

Summary Possible workarounds for ORA-1578 - Bulletin 108491.543

+=+

See bulletin 108491.543.

+=+=+

References

I/O Error Reading Block

1 Prob# 1013621.6 ORA-1115 I/O ERROR READING BLOCK

2 Soln# 2061743.6 SOLVING ORA-1115

Prob# 1013621.6 ORA-1115 I/O ERROR READING BLOCK

Problem ID: 1013621.6
Affected Platforms: Generic: not platform specific
Affected Products: Oracle7 Server
Affected Components: RDBMS Generic
Affected Oracle Vsn: Generic

Summary

ORA-1115 I/O ERROR READING BLOCK

Problem Description An ORA-1115 is issued whenever the Oracle server is unable to read from an open data file because of an I/O error:

```
01115, 00000, "IO error reading block from file %s (block # %s)"
// *Cause: Device on which the file resides is probably offline
// *Action: Restore access to the device
```

ORA-1115s are usually followed by:

- ORA-1110
- An operating system-level Oracle error message such as ORA-737X
- An operating system error (such as error number 5 in UNIX)

Problem Explanation

What causes ORA-1115?

The Oracle server delivers read-from-file requests to the underlying operating system (except if raw devices are being used). A read request specifies a data file and a block number to be accessed. If a low-level I/O error prevents the read from completing successfully, the Oracle server signals an ORA-1115.

The main causes for an ORA-1115 are:

1 Hardware problems

- Disk controller problems (the most common, and usually intermittent)
- Disk problems (including bad blocks and disk malfunctioning)

2 Data block corruption (at the physical level)

Usually caused by previous hardware problems.

3 Problems handling very large data files

In Oracle 7.1.4 and lower on Sun Solaris, bug 233569 causes ORA-1115 and ORA-7371 when handling data files bigger than 2GB.

Typical scenarios where ORA-1115 can happen include:

- On execution of DML statements
- During exports or imports
- At startup or shutdown

Diagnostics and References

Soln# 2061743.6 SOLVING ORA-1115

Solution ID: 2061743.6
For Problem: 1013621.6
Affected Platforms: Generic: not platform specific
Affected Products: Oracle7 Server
Affected Components: RDBMS Generic
Affected Oracle Vsn: Generic

Summary Solving ORA-1115

+=+

Solution Description Because most ORA-1115s are caused by hardware problems, the solution consists in first isolating the hardware problems, and then addressing the problem at the database level, if necessary.

Performing hardware checks is essential. If hardware problems are not fixed, trying to solve the problem at the database level will be useless. Run operating system level utilities and diagnostic tools that check for the sanity of disks, controllers, and the I/O subsystem. Pay special attention to the disk where the data file referenced in the ORA-1115 resides. Your system administrator should be able to assist you in this task. Such diagnostics should be done in parallel with the steps recommended here, if feasible, or as soon as possible thereafter.

Determining the exact cause of an ORA-1115 is not always trivial. Approaches differ according to whether you know the cause of the problem or not.

Steps for Solving the Problem When the Cause Is Not Known

- 1** Try to assess the cause and extent of the problem.

Examine the `alert.log` file for this instance. Scan the last few days' entries for other occurrences of ORA-1115. If you find them, determine the following:

- a** Do they reference files in different disks?
If so, it is likely that there you have controller problems. Go to Scenario II.A below.
- b** Do they reference different files in the same disk?
If so, it is likely that there are problems with that disk. Go to Scenario II. B below.
- c** Do they always reference the same data file?
If so, it is likely that the data file contains bad blocks. Go to Scenario II.C below.
If the file is bigger than 2GB and you are running 7.1.4 or lower on Solaris platform, see Scenario II.D below.
- d** If none of the above apply, go to step 2.

- 2** If the data file is in the SYSTEM tablespace, or the database is in NOARCHIVELOG mode, shut the database down. Go to step 4.

If shutdown immediate fails, do a shutdown abort.

- 3** If the database is in ARCHIVELOG mode, you should still shut the database down. If the database cannot be shut down, take the data file offline.

```
ALTER DATABASE DATAFILE '<full_path_file_name>' OFFLINE;
```

- 4** Try to copy the data file to another disk (managed by a different controller, if possible).

- 5 If the copy fails, even after you retry, the data file must be considered lost at this point. The next action depends on the tablespace to which the lost file belongs. See the following Solution References to PR entries, according to the different types of tablespaces, for instructions on how to proceed.

Important: While going through the PR entries below, keep in mind that if you restore the data file from backup, you need to place it in another disk, preferably under a different controller, and rename it inside the Oracle server (see the solution Reference to PR entry 1013480.6 for details). If you recreate any tablespace, make sure its data files are created in another disk, preferably under a different controller.

TABLESPACE	PR ENTRY
-----	-----
system	1013182.6
rollback	1013221.6
user	1013173.6
index	1013115.6
temporary	1013104.6
read-only	1013129.6

- 6 If the database is down, mount it.

- 7 Rename the data file that you succeeded in copying inside Oracle.

```
ALTER DATABASE RENAME FILE '<old_full_path_file_name>'
TO '<new_full_path_file_name>';
```

- 8 If the database is mounted, open it. If you took the data file offline, perform media recovery on it, and then bring it online.

```
RECOVER DATAFILE '<full_path_file_name>';
ALTER DATABASE DATAFILE '<full_path_file_name>' ONLINE;
```

Steps for Solving the Problem When the Cause Is Known

Controller Problems These are typically intermittent. Usually, there is no damage to the data files. Unless you can quickly fix the controller and restore access to the data file, follow these steps:

- 1 Find out which data files are under the bad controller.

Query V\$DATAFILE to obtain the names of all data files in the database. You may need the help of the system administrator to determine which data files reside in disks managed by this controller.

- 2 If any of the data files under the bad controller belongs to the SYSTEM tablespace, or if the database is in NOARCHIVELOG mode, shut the database down. Go to step 4.

If shutdown immediate fails, do a shutdown abort.

- 3 If the database is in ARCHIVELOG mode and none of the data files under the bad controller are in the SYSTEM tablespace, you should shut the database down. If the database cannot be shut down, take all the data files under the bad controller offline.

```
ALTER DATABASE DATAFILE '<full_path_file_name>' OFFLINE;
```

- 4 Try to copy all the data files under the bad controller to disks managed by different controllers.

- 5 If the database is down, mount it.

- 6 Rename all the files that you succeeded in copying inside the Oracle server.

```
ALTER DATABASE RENAME FILE '<old_full_path_file_name>'
TO '<new_full_path_file_name>';
```

- 7 If the copy fails for one or more of the data files even after you try to copy them, those data files must be considered lost at this point. See the following Solution References to PR entries, according to the tablespaces to which the lost data files belong, for instructions on how to proceed.

Important: While going through the PR entries below, keep in mind that if you restore data files from backup, you must place them in disks under other controllers and rename them inside the Oracle Server (see the solution Reference to PR entry 1013480.6 for details). If you recreate any tablespace, make sure its data files are created under other controllers.

TABSPACE	PR ENTRY
-----	-----
system	1013182.6
rollback	1013221.6
user	1013173.6
index	1013115.6
temporary	1013104.6
read-only	1013129.6

- 8** If the database is mounted, open it. If any of the moved data files is offline, apply media recovery to it, and then put it online:

```
RECOVER DATAFILE '<full_path_file_name>';  
ALTER DATABASE DATAFILE '<full_path_file_name>' ONLINE;
```

Disk Problems If a disk has bad blocks or is malfunctioning, you should focus on moving its data files to a different disk, if possible. If not, you must consider the files lost and address the issue according to the tablespaces to which they belong, while you fix the disk. The steps to follow in this scenario are like those in Scenario II.A above.

Data Block Corruption If you are certain that the data file has bad blocks, the data file should be considered LOST if it belongs to the SYSTEM tablespace or to a ROLLBACK or READ-ONLY tablespace. See the following Solution References to PR entries, depending on the tablespace to which the data file belongs.

Important: While going through the PR entries below, keep in mind that if you restore data files from backup, you must place them in different disks (preferably under other controllers) and rename them inside the Oracle server (see the solution Reference to PR entry 1013480.6 for details). If you re-create any tablespace, make sure its data files are created on different disks (preferably under other controllers).

TABSPACE	PR ENTRY
-----	-----
system	1013182.6
rollback	1013221.6
user	1013173.6
index	1013115.6
temporary	1013104.6
read-only	1013129.6

If the data file belongs to a user or index tablespace, you may also address the problem as an object re-creation issue if the ORA-1115 occurs consistently against the same objects (tables, indexes, and so on.). The following query returns the object in which the bad block is:

```
SELECT SEGMENT_NAME, SEGMENT_TYPE FROM DBA_EXTENTS
WHERE FILE_ID = <file_number> and <block_number> BETWEEN BLOCK_ID
AND BLOCK_ID + BLOCKS - 1;
```

where *<file_number>* and *<block_number>* are those listed in the ORA-1115. If this query consistently points to a table or index, you may try re-creating them, if possible, in a different tablespace. For further details on this scenario, see the Solution Reference to PR entry 1010640.6.

Very Large Data File Problems on Solaris If you are running Oracle 7.1.4 or lower on a Solaris platform, and you get an ORA-7371 with the ORA-1115 and the file is bigger than 2GB, you are likely experiencing bug 233569. This bug is fixed in 7.1.6, and patches are available for 7.1.3 (bug 233569) and 7.1.4 (bug 281904).

Rollback Segment Needs Recovery

Prob# 1010700.6 BULLETIN: ROLLBACK SEGMENT NEEDS RECOVERY

Problem ID: 1010700.6
Affected Platforms: Generic: not platform specific
Affected Products: Oracle7 Server
Affected Components: RDBMS Generic
Affected Oracle Vsn: Generic

Summary Bulletin: Rollback segment needs recovery

Problem Description

Document ID: 107693.969
Title: ROLLBACK SEGMENT NEEDS RECOVERY
Department: RDBMS SUPPORT
Creation Date: 13-February-1995
Last Revision Date: 7-June-1995
Expiration Date:
Revision Number: 1
Distribution Code: EXTERNAL
Category:
Product: GENERIC
Product Version: GENERIC
Platform: GENERIC
Information Type: ADVISORY
Impact: MEDIUM
Abstract: This article discusses what it means when a rollback segment needs recovery and how to resolve it.
Keywords: ROLLBACK;SEGMENT;NEEDS;RECOVERY;STATUS;CORRUPT

Overview

This bulletin discusses why a rollback segment has the status of “needs recovery,” what the status means, and how to resolve it.

Introduction

Rollback segments can be monitored through the data dictionary view, `DBA_ROLLBACK_SEGS`. There is a status column that describes what state the rollback segment is currently in. Normal states are either online or offline. Occasionally, the status of needs recovery appears.

When a rollback segment is in this state, bringing the rollback segment offline or online either through the `alter rollback segment` command, or by removing it from the `ROLLBACK_SEGMENTS` parameter in the `init.ora`, usually has no effect.

Understanding

A rollback segment falls into this status of needs recovery whenever the Oracle server tries to roll back an uncommitted transaction in its transaction table and fails.

Here are some examples of why a transaction may need to rollback:

- 1 A user may do a DML transaction and decide to issue rollback.
- 2 A shutdown abort occurs, and the database must do an instance recovery, in which case, the Oracle server has to roll back all uncommitted transactions.

When a rollback of a transaction occurs, undo must be applied to the data block in which the modified rows are found. If that data block is unavailable, the undo cannot be applied. The result is a corrupted rollback segment with the status of needs recovery.

What could be some reasons a data block is inaccessible for undo?

- 1 If a tablespace or a data file is offline or missing
- 2 If the object the data block belongs to is corrupted
- 3 If the data block that is corrupt is in the rollback segment itself rather than the object

How to Resolve the Needs Recovery Status

- 1 Verify that all tablespaces and all data files are online. This can be checked through V\$DATAFILE, under the STATUS column. For tablespaces associated with the data files, look in DBA_TABLESPACES.
If that still does not resolve the problem, continue with the following steps.
- 2 Put the following in the `init.ora`- `event = "10015 trace name context forever, level 10"`.
Setting this event will generate a trace file that reveals the necessary information about the transaction that the Oracle server is trying to roll back and most importantly, what object the Oracle server is trying to apply the undo to.
- 3 Shut down the database (if the NORMAL mode does not work, try IMMEDIATE or ABORT) and bring it back up.
Note: An ora-1545 or other errors may be encountered. If the database cannot start up, contact customer support.
- 4 Check in the directory that is specified by the USER_DUMP_DEST parameter (in the `init.ora` or SHOW Parameter command) for a trace file that was generated at startup time.
- 5 In the trace file, there should be a message similar to error recovery tx(##) object #. TX(##) refers to transaction information. The object # is the same as the object_id in sys.dba_objects.
- 6 Use the following query to determine what object the Oracle server is trying to perform recovery on.

```
select owner, object_name, object_type, status
from dba_objects where object_id = <object #>;
```
- 7 This object must be dropped so that the undo can be released. An export, or relying on a backup, may be necessary to restore the object after the corrupted rollback segment disappears.
- 8 After dropping the object, put the rollback segment back in the `init.ora` parameter ROLLBACK_SEGMENTS, remove the event, and shut down and start up the database.

In most cases, the above steps will resolve the problematic rollback segment. If this still does not resolve the problem, it may be likely that the corruption is in the actual rollback segment. If the problem is not resolved, please contact customer support.

Problem Explanation

Diagnostics and References

- * {6123.6,Y,100} ROLLBACK SEGMENT NEEDS RECOVERY
- * {6124.6,Y,100} ORA-1545 ON STARTUP

Lost Data File in Rollback Segment

Prob# 1013221.6 RECOVERING FROM A LOST DATAFILE IN A ROLLBACK TABLESPACE

Problem ID: 1013221.6
Affected Platforms: Generic: not platform specific
Affected Products: Oracle7 Server
Affected Components: RDBMS Generic
Affected Oracle Vsn: Generic

Summary Recovering from a lost data file in a rollback tablespace

Problem Description This is a recovery scenario in which a data file in a rollback segment tablespace has been lost or damaged to a point that the Oracle server cannot recognize it anymore. Trying to start up the database will result in ORA-1157, ORA-1110, and possibly an operating system level error such as ORA-7360. Trying to shut down the database in normal or immediate mode will result in ORA-1116, ORA-1110, and possibly an operating system level error such as ORA-7368.

Solution Description This recovery situation requires extra caution. Please call Oracle Customer Support if you have questions or need assistance.

The main issue in solving this problem is trying to make sure that the active transactions in the rollback segments do not get lost.

Solution Explanation The approach depends on the specific scenario in which the loss of the rollback data file is detected.

The Database Is Down

Attempting to start up the database will result in ORA-1157 and ORA-1110. The solution here depends on whether the database was cleanly shut down or not.

The Database Was Cleanly Shut Down If you are absolutely positive that the database was cleanly shut down, that is, closed with either shutdown NORMAL or IMMEDIATE, then the simplest solution is to drop the missing data file offline, open the database in restricted mode, and then drop and recreate the rollback tablespace to which the file belonged. Do not follow this procedure if the database was shut down in ABORT mode or if it crashed.

The steps are:

- 1 Make sure the database was last cleanly shut down.

Check the `alert.log` file for this instance. Go to the bottom of the file and make sure the last time you shut the database down you received the messages:

```
"alter database dismount
Completed: alter database dismount"
```

This also includes the case of a clean shutdown followed by a failed attempt to start up the database. In that case, the Oracle server will issue error messages and shut itself down abort. For the purposes of this solution, though, this counts as a clean shutdown.

If that is not the case, that is, if the last time you shut the database down, it was in ABORT mode, or the database crashed itself, it is *not* safe to proceed. You should follow the instructions for case I.B below.

- 2 Remove all the rollback segments in the tablespace to which the lost data file belongs from the `ROLLBACK_SEGMENTS` parameter in the `init.ora` file for this instance. If you are not sure about which rollbacks are in that tablespace, simply comment out the whole `ROLLBACK_SEGMENTS` entry.
- 3 Mount the database in restricted mode.

```
STARTUP RESTRICT MOUNT
```

- 4 Drop the lost data file offline.

```
ALTER DATABASE DATAFILE '<full_path_file_name>' OFFLINE DROP;
```

- 5 Open the database.

```
ALTER DATABASE OPEN
```

If you receive the message "Statement processed," go to step 7.

If instead you get ORA-604, ORA-376, and ORA-1110, go to step 6.

- 6** Because opening the database failed, shut the database down and edit the `init.ora` file for this instance.

Comment out the `ROLLBACK_SEGMENTS` parameter and add the following line:

```
_corrupted_rollback_segments = ( <rollback1>, ..., <rollbackN> )
```

For example, the above list should contain all the rollbacks originally listed in the `ROLLBACK_SEGMENTS` parameter.

Use this parameter only in this specific scenario or as instructed by Oracle Customer Support, then start up the database in restricted mode:

```
STARTUP RESTRICT
```

- 7** Drop the rollback tablespace to which the data file belonged.

```
DROP TABLESPACE <tablespace_name> INCLUDING CONTENTS;
```

- 8** Recreate the rollback tablespace with all its rollback segments. Remember to bring the rollbacks online after you create them.

- 9** Make the database available to all users.

```
ALTER SYSTEM DISABLE RESTRICTED SESSION;
```

- 10** Reininclude the rollbacks you just recreated in the `ROLLBACK_SEGMENTS` parameter in the `init.ora` file for this instance. If you had commented out the whole `ROLLBACK_SEGMENTS` entry, simply uncomment it now. If you had to go through step 6, remove the corrupted `ROLLBACK_SEGMENTS` parameter now.

The Database Was Not Cleanly Shut Down This is the situation where the database was last shut down in ABORT or CRASHED mode. In this case, it is almost certain that the rollback segments that had extents in the lost data file still contain active transactions. Therefore, the file cannot be taken offline or dropped. You must restore the lost data file from a backup and apply media recovery to it. If the database is in NOARCHIVELOG mode, you will only succeed in recovering the data file if the redo to be applied is within the range of your online logs. If a backup of the data file is not available, please contact Oracle Customer Support.

These are the steps:

- 1 Restore the lost file from a backup.
- 2 Mount the database.
- 3 Issue the following query:

```
SELECT FILE#, NAME, STATUS FROM V$DATAFILE;
```

If the status of the file you just restored is OFFLINE, you must take it online before proceeding:

```
ALTER DATABASE DATAFILE '<full_path_file_name>' ONLINE;
```

- 4 Issue the following query:

```
SELECT V1.GROUP#, MEMBER, SEQUENCE#, FIRST_CHANGE#
FROM V$LOG V1, V$LOGFILE V2
WHERE V1.GROUP# = V2.GROUP# ;
```

This will list all your online redo log files and their respective sequence and first change numbers.

- 5 If the database is in NOARCHIVELOG mode, issue the query:

```
SELECT FILE#, CHANGE# FROM V$RECOVER_FILE;
```

If the CHANGE# is greater than the minimum FIRST_CHANGE# of your logs, the data file can be recovered. Remember that all the logs to be applied will be online logs, and go to step 6.

If the CHANGE# is lesser than the minimum FIRST_CHANGE# of your logs, the file cannot be recovered. Your options at this point include restoring a full backup if one is available or forcing the database to open in an inconsistent state to get a full export out of it. For further details and to assist you in your decision, please contact Oracle Customer Support.

- 6 Recover the data file:

```
RECOVER DATAFILE '<full_path_file_name>'
```

- 7 Confirm each of the logs that you are prompted for until you receive the message “Media recovery complete.” If you are prompted for a non-existing archived log, the Oracle server probably needs one or more of the online logs to proceed with the recovery. Compare the sequence number referenced in the ORA-280 message with the sequence numbers of your online logs. Then enter the full path name of one of the members of the redo group whose sequence number matches the one you are being asked for. Continue to enter online logs as requested until you receive the message, “Media recovery complete.”
- 8 Open the database.

The Database Is Up

If you have detected the loss of the rollback data file and the database is still up and running, do not shut it down. In most cases, it is simpler to solve this problem with the database up than with it down.

Two approaches are possible in this scenario:

- 1 The first one involves taking the lost data file offline, restoring it from backup, and then applying media recovery to it to make it consistent with the rest of the database. This method can only be used if the database is in ARCHIVELOG mode.
- 2 The other approach involves taking offline all the rollback segments in the tablespace to which the lost data file belongs, dropping the tablespace, and then recreating it. You may need to kill sessions that have transactions in the rollbacks involved to force the rollbacks to go offline.

In general, the first approach is simpler to apply. It will also be faster if the data file and the necessary archived logs can be quickly restored from backup. However, more user transactions will error out and be rolled back than with the second approach. Because of read-consistency, queries against certain tables may fail with the first approach, because the rollback extents from which the Oracle server would retrieve the data may be in the offlined data file.

Approach A: Restoring the Data File from Backup As mentioned before, this approach can only be followed if the database is in ARCHIVELOG mode. Here are the steps:

- 1 Take the lost data file offline.

```
ALTER DATABASE DATAFILE '<full_path_file_name>' OFFLINE;
```

Note: Depending on the current amount of database activity, you may need to create additional rollback segments in a different tablespace to keep the database going while you take care of the problem.

- 2 Restore the data file from a backup.

- 3 Issue the following query:

```
SELECT V1.GROUP#, MEMBER, SEQUENCE#  
FROM V$LOG V1, V$LOGFILE V2  
WHERE V1.GROUP# = V2.GROUP# ;
```

This will list all your online redo log files and their respective sequence numbers.

- 4 Recover the data file:

```
RECOVER DATAFILE '<full_path_file_name>'
```

- 5 Confirm each of the logs that you are prompted for until you receive the message, “Media recovery complete.” If you are prompted for a non-existing archived log, the Oracle server probably needs one or more of the online logs to proceed with the recovery. Compare the sequence number referenced in the ORA-280 message with the sequence numbers of your online logs. Then enter the full path name of one of the members of the redo group whose sequence number matches the one you are being asked for. Continue to enter online logs as requested until you receive the message “Media recovery complete.”

- 6 Bring the data file back online.

```
ALTER DATABASE DATAFILE '<full_path_file_name>' ONLINE;
```

Approach B: Re-creating the Rollback Tablespace This approach can be used regardless of the archival mode of the database. The steps are:

- 1 Try to take offline all of the rollback segments in the tablespace to which the lost data file belongs.

```
ALTER ROLLBACK SEGMENT <rollback_segment> OFFLINE;
```

Repeat this statement for all rollbacks in the tablespace.

Note: Depending on the current amount of database activity, you may need to create additional rollback segments in a different tablespace to keep the database going while you take care of the problem.

- 2 Check the status of the rollbacks.

They must all be offline before they can be dropped. Issue the query:

```
SELECT SEGMENT_NAME, STATUS FROM DBA_ROLLBACK_SEGS
WHERE TABLESPACE_NAME = '<TABLESPACE_NAME>';
```

- 3 Drop all offlined rollback segments.

For each rollback returned by the query in step 2 with status OFFLINE, issue the statement:

```
DROP ROLLBACK SEGMENT <rollback_segment>;
```

- 4 Handle the rollbacks that remain online.

Repeat the query in step 2.

If any of the rollbacks you tried to offline still has an ONLINE status, it means there are still active transactions in it. You may confirm that by issuing the query:

```
SELECT SEGMENT_NAME, XACTS_ACTIVE_TX, V.STATUS
FROM V$ROLLSTAT V, DBA_ROLLBACK_SEGS
WHERE TABLESPACE_NAME = '<TABLESPACE_NAME>' AND SEGMENT_ID = USN;
```

If the above query returns no rows, it means all the rollbacks in the affected tablespace are already offline. Repeat the query in step 2 to retrieve the names of the rollbacks that just became offline and then drop them as described in step 3.

If the above query returns one or more rows, they should show the status of PENDING OFFLINE.

Next, check the ACTIVE_TX column for each rollback. A value of 0 implies that there are no pending transactions left in the rollback, and it should go offline shortly. Repeat the query in step 2 a few more times until it shows the rollback being offline, and then drop it as described in step 3. Go to step 6.

If any of the PENDING OFFLINE rollbacks has a value of 1 or greater in the ACTIVE_TX column, go to step 5.

5 Force rollbacks with active transactions to go offline.

At this point, the only way to move forward is to have the PENDING OFFLINE rollbacks released. The active transactions in these rollbacks must either be committed or rolled back. The following query shows which users have transactions assigned to which rollbacks:

```
SELECT S.SID, S.SERIAL#, S.USERNAME, R.NAME "ROLLBACK"
FROM V$SESSION S, V$TRANSACTION T, V$ROLLNAME R
WHERE R.NAME IN ( '<PENDING_ROLLBACK_1>', ... ,
'<PENDING_ROLLBACK_N>' )
AND S.TADDR = T.ADDR AND T.XIDUSN = R.USN;
```

You may directly contact the users with transactions in the PENDING OFFLINE rollbacks and ask them to commit (preferably) or rollback immediately. If that is not feasible, you can force that to happen by killing their sessions. For each of the entries returned by the above query, issue the statement:

```
ALTER SYSTEM KILL SESSION '<SID>, <SERIAL#>';
```

Where *<SID>* and *<SERIAL#>* are those returned by the previous query. After the sessions are killed, it may take a few minutes before the Oracle server finishes rolling back and doing cleanup work. Go back to step 2 and repeat the query periodically until all rollbacks in the affected tablespace are offline and ready to be dropped.

6 Drop the rollback tablespace.

```
DROP TABLESPACE <tablespace_name> INCLUDING CONTENTS;
```

If this statement fails, please contact Oracle Customer Support. Otherwise, proceed to step 7.

7 Re-create the rollback tablespace.

8 Re-create the rollback segments in the tablespace and bring them online.

- 1 Prob# 1012943.6 ORA-1113 FILE NEEDS MEDIA RECOVERY
- 2 Soln# 2061115.6 PERFORM MEDIA RECOVERY ON THE DATAFILE(S)

Prob# 1012943.6 ORA-1113 FILE NEEDS MEDIA RECOVERY

Problem ID: 1012943.6
 Affected Platforms: Generic: not platform specific
 Affected Products: Oracle7 Server
 Affected Components: RDBMS Generic
 Affected Oracle Vsn: Generic

Summary ORA-1113 file needs media recovery

Problem Description An ORA-1113 will be issued whenever a data file is not in sync with the rest of the database: 01113, 00000, "file %s needs media recovery"

```
// *Cause: An attempt was made to online or open a database with a
file that
//      is in need of media recovery.
// *Action: First apply media recovery to the file.
```

Often, ORA-1113 occurs together with ORA-1110. The most common scenarios for an ORA-1113 are:

- 1 At startup time (usually followed by ORA-1110)
 - The database crashed or was shut down in ABORT mode, or the machine was rebooted while the data file's tablespace was in hot backup mode. At startup, you get ORA-1113.
 - You attempt to open the database with an old version of a data file that was restored from a backup without first bringing it up-to-date.
- 2 Trying to online a data file

You try to bring an offline data file back online and get ORA-1113.

Problem Explanation Oracle's architecture is tightly coupled in the sense that all database files, data files, redo log files, and control files must be in sync when the database is opened or at the end of a checkpoint. This implies that the checkpoint System Commit Number (SCN) of all data files must be the same. If that is not the case for a particular data file, an ORA-1113 will be generated. For example, when you put a tablespace in hot backup mode, the checkpoint SCN of all its data files is frozen at the current value until you issue the corresponding end backup. If the database crashes during a hot backup and you try to restart it without doing recovery, you will likely get ORA-1113 for at least one of the data files in the tablespace that was being backed up, because its SCN will probably be lower than that of the control file and the data files in other tablespaces. Likewise, offlining a data file causes its checkpoint SCN to freeze. If you simply attempt to take the file online without recovering it first, its SCN will likely be much older than that of the online data files, and thus an ORA-1113 will result.

Diagnostics and References

- * {6687.6,Y,100} IS THIS HAPPENING ON STARTUP?
- * {6690.6,Y,100} DID YOU TRY AND ONLINE THE DATAFILE?
- * {7241.6,Y,100} BRINGING A TABLESPACE ONLINE

Soln# 2061115.6 PERFORM MEDIA RECOVERY ON THE DATAFILE(S)

Solution ID: 2061115.6
For Problem: 012943.6
Affected Platforms: Generic: not platform specific
Affected Products: Oracle7 Server
Affected Components: RDBMS Generic
Affected Oracle Vsn: Generic

Summary Perform media recovery on the data files

Solution Description The solution for an ORA-1113 is to perform media recovery on the files having problems using the RECOVER DATAFILE command. If you know that most or all of the files in a tablespace need to be recovered and the database is open, use RECOVER TABLESPACE. If a number of tablespaces need recovery, use RECOVER DATABASE with the database mounted.

The way to do that varies slightly according to the scenario.

Solution Explanation Start by querying V\$LOG and V\$LOGFILE. If the database is down, you must mount it first. Connect internal in SQL*DBA or Server Manager and issue the query:

```
SELECT V1.GROUP#, MEMBER, SEQUENCE#, FIRST_CHANGE#  
FROM V$LOG V1, V$LOGFILE V2  
WHERE V1.GROUP# = V2.GROUP# ;
```

This will list all your online redo log files and their respective sequence and first change numbers.

The steps to take next depend on the scenario in which the ORA-1113 was issued:

1 At startup after crash with tablespaces in hot backup

a With Oracle 7.1 or lower

- i. Mount the database.
- ii. Apply media recovery to the database.

```
RECOVER DATABASE
```

- iii. Confirm each of the archived logs that you are prompted for until you receive the message, "Media recovery complete." If you are prompted for an archived log that does not exist, the Oracle server probably needs one or more of the online logs to proceed with the recovery. Compare the sequence number referenced in the ORA-280 message with the sequence numbers of your online logs. Then enter the full path name of one of the members of the redo group whose sequence number matches the one you are being asked for. Keep entering online logs as requested until you receive the message "Media recovery complete."
- iv. Open the database.

b With Oracle 7.2 or higher

- i. Mount the database.
- ii. Find out which data files were in hot backup mode when the database crashed or was shut down in ABORT mode, or the machine was rebooted by running the query:

```
SELECT V1.FILE#, NAME
FROM V$BACKUP V1, V$DATAFILE V2
WHERE V1.STATUS = 'ACTIVE' AND V1.FILE# = V2.FILE# ;
```

- iii. For each of the files returned by the above query, issue the command:

```
ALTER DATABASE DATAFILE '<full path name>' END BACKUP;
```
- iv. Open the database.

2 At startup after restoring a data file or tablespace from a backup

a With the database in ARCHIVELOG mode

- i. Mount the database.
- ii. Recover the data file:

```
RECOVER DATAFILE '<full path name>'
```

 If recovering more than one data file, issue the following:

```
RECOVER DATABASE
```

iii. Confirm each of the archived logs that you are prompted for until you receive the message, "Media recovery complete." If you are prompted for an archived log that does not exist, the Oracle server probably needs one or more of the online logs to proceed with the recovery. Compare the sequence number referenced in the ORA-280 message with the sequence numbers of your online logs. Then enter the full path name of one of the members of the redo group whose sequence number matches the one you are being asked for. Keep entering online logs as requested until you receive the message, "Media recovery complete."

iv. Open the database.

b With the database in NOARCHIVELOG mode

In this case, you will only succeed in recovering the data file or tablespace if the redo to be applied to it is within the range of your online logs. Issue the query:

```
SELECT FILE#, CHANGE# FROM V$RECOVER_FILE;
```

Compare the change number you obtain with the FIRST_CHANGE# of your online logs.

If the CHANGE# is greater than the minimum FIRST_CHANGE# of your logs, the data file can be recovered. In this case, the procedure to be followed is like that of scenario II.A above, except that you must always enter the appropriate online log when prompted, until recovery is finished.

If the CHANGE# is lesser than the minimum FIRST_CHANGE# of your logs, the file cannot be recovered. Your options at this point include:

- If the data file is in a temporary or index tablespace, you may drop it with an `ALTER DATABASE DATAFILE '<full path name>' OFFLINE DROP` statement and then open the database. Once the database is up, you must drop the tablespace to which the data file belongs and re-create it.
- If the data file is in the SYSTEM or in a rollback tablespace, restore an up-to-date copy of the data file (if available) or your most recent full backup. If a full, consistent backup is not available, please contact Oracle Customer Support.
- For all other cases in this scenario, you must weigh the cost of going to a backup versus the cost of recreating the tablespace involved, as described in the two previous cases. For more details or to assist you in your decision, please contact Oracle Customer Support.

3 Trying to online a data file or tablespace

a Recover the data file:

```
RECOVER DATAFILE '<full path name>'
```

If recovering a tablespace, do the following:

```
RECOVER TABLESPACE <tablespace>
```

- b** Confirm each of the archived logs that you are prompted for until you receive the message, “Media recovery complete.” If you are prompted for an archived log that does not exist, the Oracle server probably needs one or more of the online logs to proceed with the recovery. Compare the sequence number referenced in the ORA-280 message with the sequence numbers of your online logs. Enter the full path name of one of the members of the redo group whose sequence number matches the one you are being asked for. Continue to enter online logs as requested until you receive the message, “Media recovery complete.”

